

Deploy WinForms on the Web

www.visualstudiomagazine.com

HOT .NET TIPS
go to visualstudiomagazine.com

Visual Studio® Magazine

A FAWCETTE TECHNICAL PUBLICATION

September 1, 2002

VOL. 12, NO. 10

DEBUG .NET APPS

Modify Table Schemas
Programmatically

Build Boilerplate
Templates

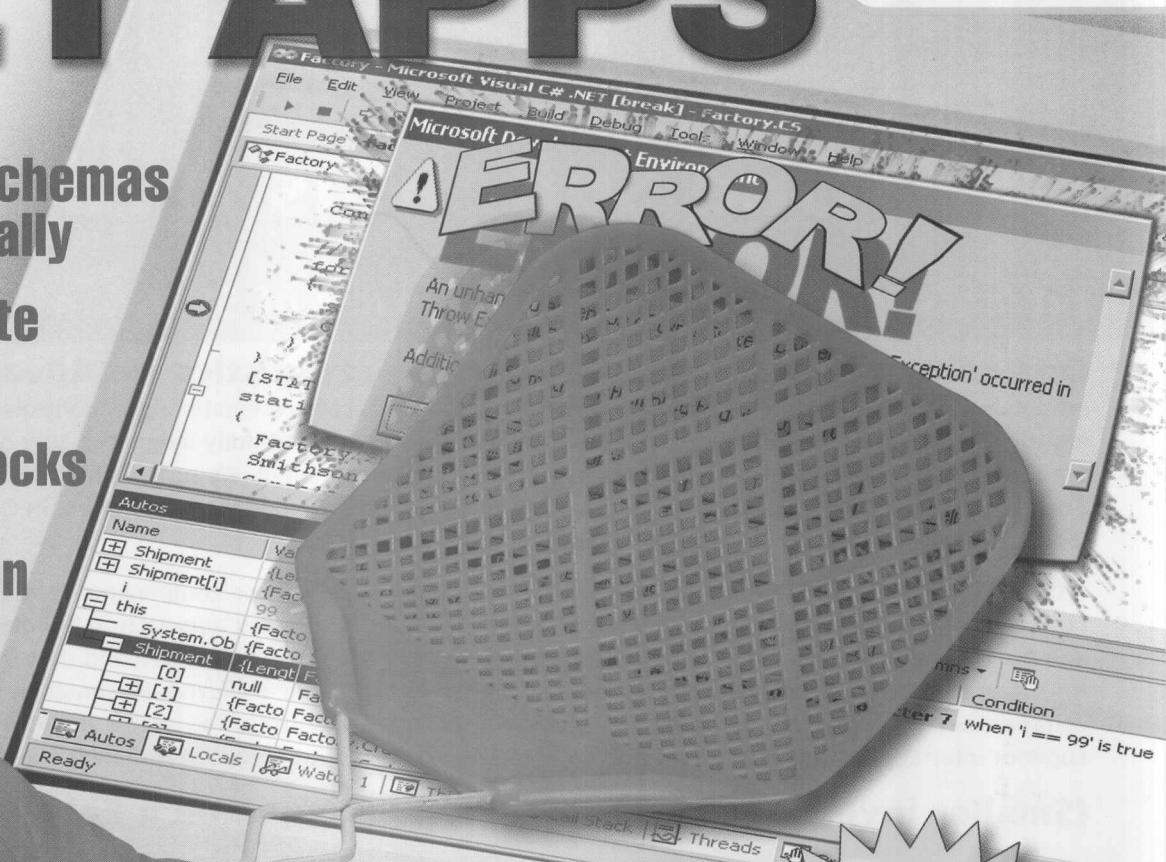
Prevent Deadlocks
With Thread
Synchronization

INCORPORATING

VISUAL BASIC
PROGRAMMER'S JOURNAL

Visual C++
Developers Journal

SQL PRO



U.S. \$6.95 (CANADA \$8.95)



Locate all
past articles
in our new
index



Only One Suite Has It All, & More

Infragistics - formed by the merger of Sheridan and ProtoView

One comprehensive suite offering -- Infragistics NetAdvantage Suite 1.2

Infragistics proudly presents an unprecedented range and depth of presentation-layer controls—UltraSuite, UltraWinSuite and UltraWebSuite—in one offering. The NetAdvantage Suite is the ONLY fully integrated suite you'll ever need to create the most flexible, advanced applications for any Microsoft environment. We provide high-performance, superior tools; Grid, Scheduling, Charting, Toolbars, Menus, Tree, UI and Editing Components and more, in COM, Windows Forms (for Visual Studio.NET) and Web Forms (for ASP.NET).

Tomorrow's architecture, today. We've planned our components carefully, applying Infragistics' famous Best Practices methodology for highest reliability. Our architected approach to building components is elegant and mature; delivering shared assemblies, a unified UI framework, reusable objects, and transferrable knowledge gained from our consistent object models. Shared assemblies (such as our exclusive Presentation Layer Framework) mean common behavior and programmability across more efficient controls. Yet, we've maintained maximum flexibility, superior interface design, database compatibility and advanced out-of-box conveniences such as RAD designers.

(Smaller investment + Bigger Suite) – Activation = Easy Math

Infragistics Products

| | |
|---|-------------------------------|
| NetAdvantage Suite v1.2 | \$ 495 |
| NetAdvantage Suite v1.2 with annual subscription | \$ 695 (includes source code) |
| NetAdvantage Suite v1.2 Enterprise Edition | \$ 995 (includes source code) |

Special Offer!

The screenshot displays a complex Windows application interface for 'Infragistics UltraWinSuite'. On the left, there's a tree view of car manufacturers and models. In the center, a grid shows vehicle details like Model, Color, MSRP, Invoice, InStock, SKU, Type, and Options. A chart is overlaid on the grid. Below the grid, a calendar shows dates from February 1998 to November 2001. To the right, a schedule window lists events such as 'Test drive BMW 325' and 'Review financial options'.

- ③ UltraWebListBars
- ④ UltraWebNavigatorTree
- ⑤ UltraWebTabs
- ⑥ NEW UltraWebGrid
- ⑦ NEW UltraWebChart

- UltraWinSuite**
- ⑧ UltraWinListBar
- ⑨ UltraWinTree
- ⑩ UltraWinGrid
- ⑪ UltraWeekView
- ⑫ UltraMonthViewMulti
- ⑬ UltraDayView
- ⑭ NEW UltraWinChart

- Infragistics Presentation Layer Framework**
- Sampling of fully customizable UIElements
- ① ItemUIElement
- ② ItemImageUIElement
- ③ GroupByRowUIElement
- ④ GroupByButtonUIElement
- ⑤ DayofWeekUIElement
- ⑥ WeekUIElement
- ⑦ NodeImageUIElement
- ⑧ ExpansionUIElement
- ⑨ RowUIElement
- ⑩ DayNumberUIElement
- ⑪ AppointmentUIElement

We've Added WebGrid & Charting!

Infragistics NetAdvantage™ Suite 1.2 includes the following:

UltraWinSuite™ 1.1

Windows Forms
Rich Client Components for Visual Studio .NET

- UltraWinGrid™
- UltraWinTree™
- UltraWinSchedule™
- UltraWinListBar™
- NEW UltraWinChart™
- UltraWinToolBars™*

UltraWebSuite™ 2

Web Forms Components for ASP.NET

- UltraWebBars™
- UltraWebNavigator™
- UltraWebGrid™
- NEW UltraWebChart™

*scheduled for version 2 release. Purchase NetAdvantage Suite 1.2 with Subscription or Enterprise, and receive version 2 automatically, for FREE

UltraSuite™ 3

ActiveX Components

- UltraGrid™
- UltraToolBars™
- ActiveTreeView™
- ScheduleX™
- DataExplorer™

NetAdvantage Suite is available with optional subscription or Enterprise Editions.

Maximize your efficiency. Minimize your code. Sacrifice nothing.



Visit www.infragistics.com
 Read the Presentation Layer
 Framework white paper
www.infragistics.com/netwinforms
 Download Free Trial Version!
 Order Online! or call 800-231-8588

Copyright 2002 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, the Presentation Layer Framework logo, NetAdvantage, UltraSuite, UltraWinSuite, UltraWinGrid, UltraWinTree, UltraWinSchedule, UltraWinToolBars, UltraWinListBar, UltraWinChart, UltraWebSuite, UltraWebBars, UltraWebNavigator, UltraWebGrid, UltraWebChart, Data Explorer, ActiveTreeView, UltraGrid, UltraToolBars, ScheduleX are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

Infragistics
 Component powered infrastructure

800-231-8588 • infragistics.com



Software FX

Any Chart, Anywhere!

Chart FX for .NET

Rewritten in C# to exist within .NET as managed code.

Now, your entire
.NET charting solution...
all in one package!

Chart FX for .NET Web Forms

- 1 Production Server license
- Unlimited CPUs per license
- 1 Development Server license

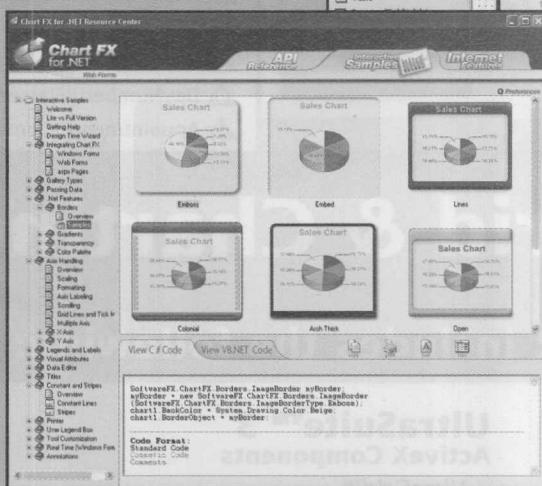
also includes...

Chart FX for .NET Windows Forms*

- 1 Server license

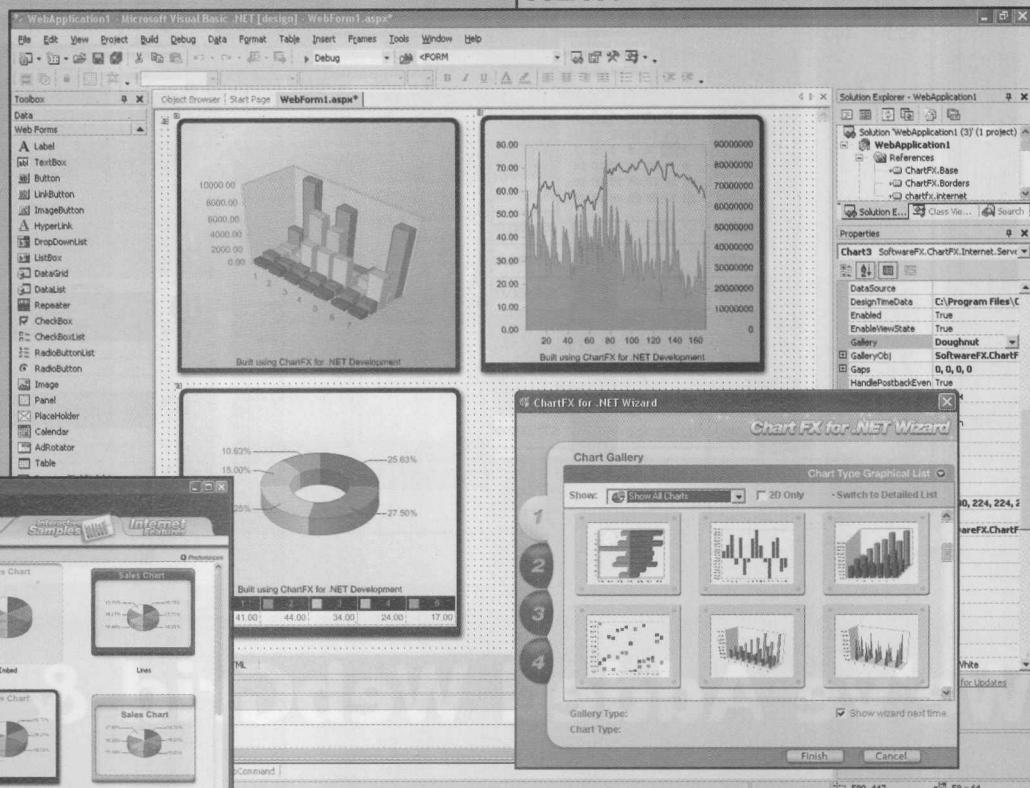
* Chart FX for .NET is also available in a Windows Forms only package.

**See web site
for pricing.**



100%
MANAGED!

NEW! Chart FX for Web Matrix
NEW! Basic charting for the Microsoft
NEW! Web Matrix development tool.



Resource Center

An interactive programmer's guide that integrates a comprehensive explanation of the different features and hundreds of samples in different languages, including C# and VB.NET.

Wizard

The Wizard will appear, when a chart is dropped into a Windows or Web Form, and assist in creating visually stimulating, useful charts before writing a single line of code.

Chart FX Lite

FREE .NET Charting

Get the FREE charting solution for Windows Forms created with Visual Studio.net. Chart FX Lite offers valuable charting functionality, including royalty-free distribution; 2D & 3D charts; grid line & axis manipulation; ability to pass data through the API or DataSets; forward code compatibility for easy migration to other Chart FX products.

**Download
Today!**



softwarefx.com

• Buy Online!

• Not Ready to Buy?

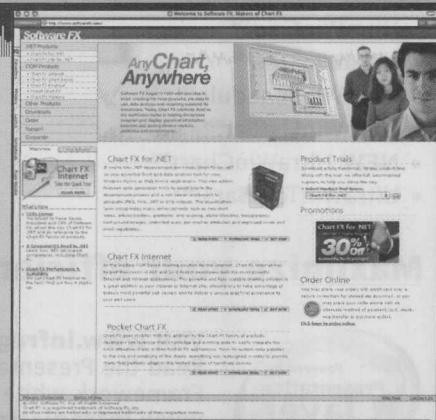
Download fully-functional trial versions.

• Catalogs and Demo CDs!

Request a catalog or demo CD be shipped to you in the U.S.—absolutely FREE!

• Quality Support!

support.softwarefx.com offers FREE online access to hundreds of Knowledge Base articles, FAQs, newsgroups and product patches and updates downloads. Support subscriptions available as well.



(561) 999-8888 • www.softwarefx.com • (800) 392-4278

Try it Today, for Free!

Download a Trial at www.softwarefx.com
or call (561) 999-8888 for your demo CD.

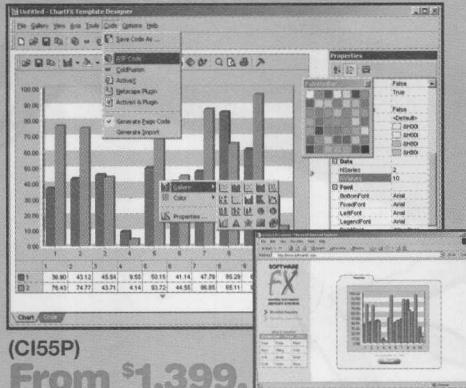


Chart FX Internet 5.5

The most advanced charting solution for Internet developers.

Chart FX Internet powerful features:

- Interactive Image Map Generation
- Drill-Down Capability
- Object Oriented API
- Auto Browser Detection & Response
- User Interface for Edit and Customization
- Bit-Streaming Directly to a Browser (required for web farms)
- ActiveX and Netscape Plug-Ins Included
- Dynamic Template and Code Generation (ASP, ColdFusion or Client-Side Code)
- Included DTC (Visual InterDev 6.0 compliant)
- Embedded Data Binding
- Highly Extensible (Annotation, CrossTab, XML)
- Included Sample Code & KnowledgeBase
- Included Wireless Extension
- Scalable Vector Graphics Generation



(CI55P) From \$1,399.

NEW VERSION!

Chart FX Client Server 5.1

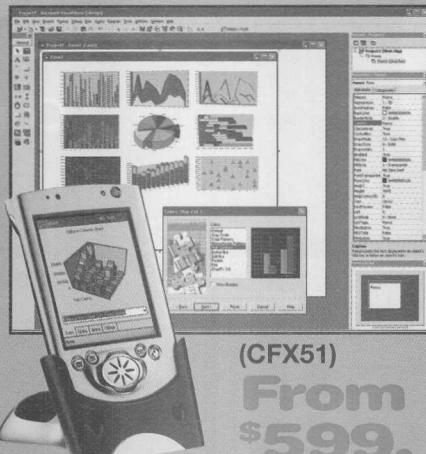
The industry leading charting tool - now embedded!

The most feature-rich client server architecture charting component available:

- Convenient Wizard Setup
- User Interface for Easy Control
- Easily Accessible API
- Active Document Server (OLE Server)
- Dynamic Template and Code Generation
- Integrated Annotation Extension
- Included Sample Code & KnowledgeBase

Pocket Chart FX Included!

- Familiar Chart FX API
- Works with eVB and eVC++
- Pre-compiled formats (MIPS, ARM, SH3/4)
- ADO-CE support
- Customizable stylus events
- Most 2D & 3D Chart FX charts supported
- Less than 400K memory footprint



(CFX51)

From \$599.

NEW VERSION!

In the works...

Chart FX Real-Time

Embedded real-time capabilities for Internet/intranet applications.

Chart FX Maps

Dynamic, data-driven image generation for .NET developers.

Chart FX for Java

Chart FX for the J2EE application server environment.

Sign Up for Beta Now!

Image Toppings

Dynamic, data-driven image generation.

(ITSRV) From \$649.

WebBar FX & WebTree FX

Enhanced information architecture & organization.

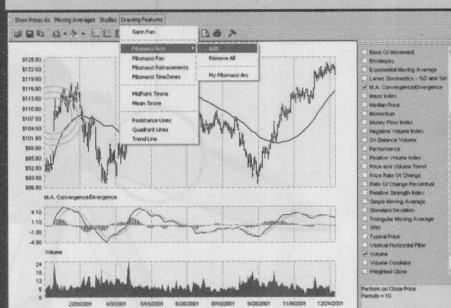
(WTFX or WBFX) \$149 ea.

(561) 999-8888 • www.softwarefx.com • (800) 392-4278

All other brand names are the property of their respective owners.

Chart FX Financial

Powerful Technical Analysis charting for financial providers.

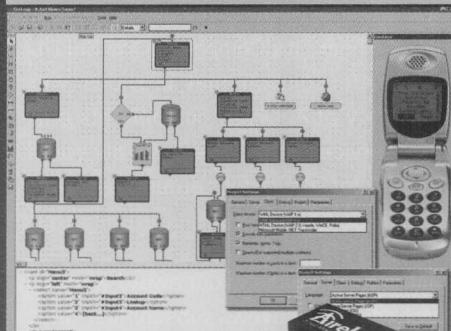


Some of the financial features are:

- 4 New Japanese Charts
- 50+ Studies to enhance chart analysis
- Up to 18 user defined indicators, calculated by the programmer and displayed as a technical or price indicator
- Up to 3 Technical Indicators in a single screen
- 12 Custom Interactive Drawing features (i.e. Fibonacci Fans, Arcs, etc.)

(CFXFE) Call for pricing.

Chart FX Wireless



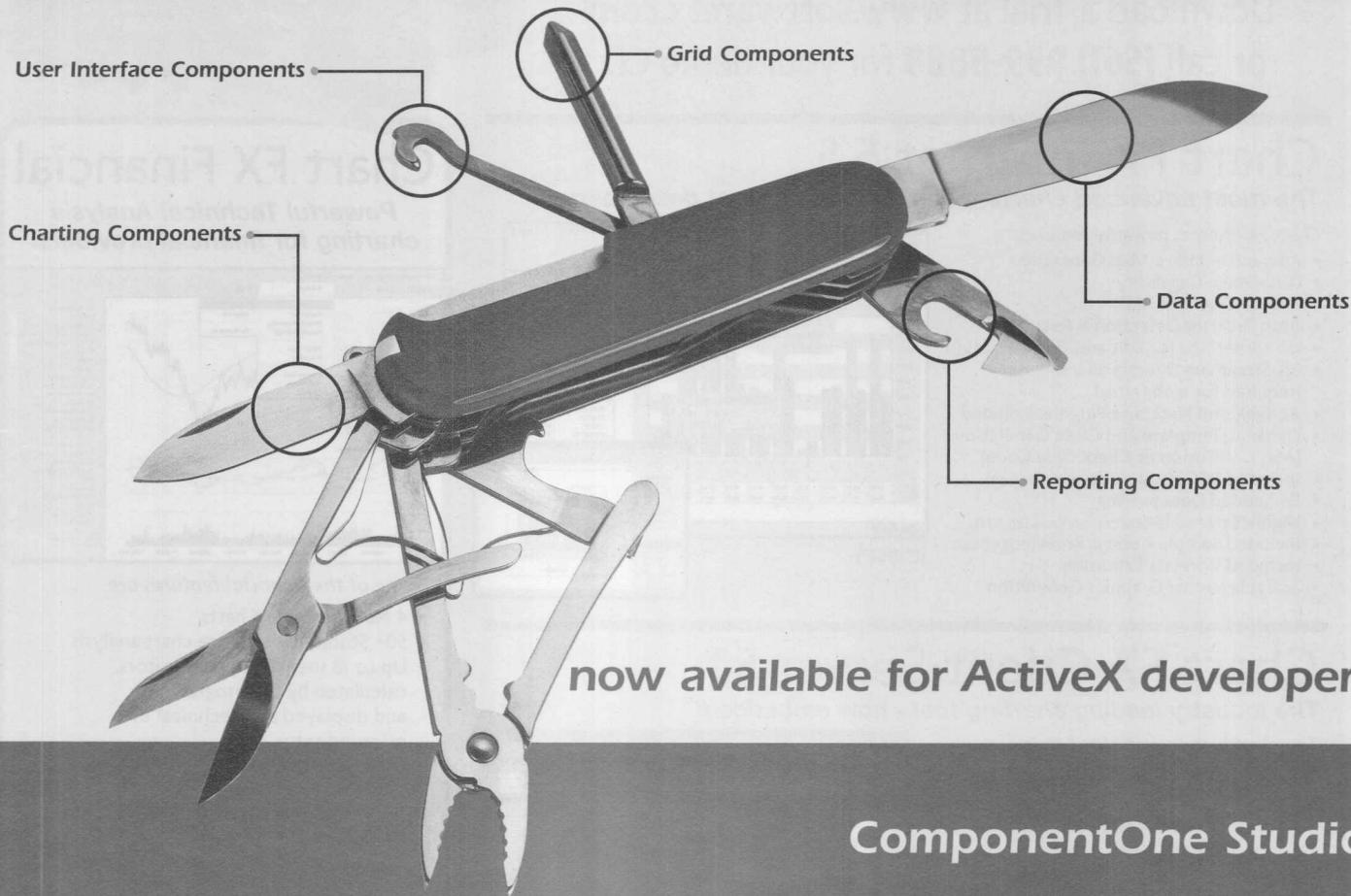
Develop robust wireless apps with powerful charting capabilities.

(CIE55ALP2)

Promo \$2,498.



The ultimate survival tool...



now available for ActiveX developers.

ComponentOne Studio™

Every developer dreams of that one tool that you can't live without. You crave the tool that will fill all of your development needs. It's real. It's here. It's ComponentOne Studio. The complete solution that provides you with a variety of ActiveX® components and support services that give you all you'll ever need for powerful, enterprise wide application development.

ComponentOne Studio is our award-winning comprehensive ActiveX subscription that provides unsurpassed business value by delivering the latest ActiveX components, updates and upgrades, new software offerings, and e-mail and online support for one full year from your date of purchase. ComponentOne Studio is the absolute best way to get your tools of choice for Microsoft® Visual Studio® development.

The heart of ComponentOne Studio is its set of top-notch, award-winning components that ComponentOne is legendary for. Below is a list of the components you'll receive with our "ultimate survival tool".

Grid Components

- VSFlexGrid® Pro 7.0
- True DBGrid® Pro 7.0

Reporting Components

- VSVIEW® 7.0 Classic Edition
- VSVIEW® 7.0 Reporting Edition

Charting Components

- ComponentOne Chart™ 7.0
- ComponentOne WebChart™ 7.0

Data Components

- ComponentOne Query™ 1.0
- ComponentOne True DBList™ Pro 7.0
- True DBInput™ Pro 6.0
- True DataControl™ 6.0

User Interface Components

- SizerOne™ 7.0
- VSFORUM™ 2.0
- VSSPELL™ 6.0 International Edition

Subscription Benefits

- **Free updates and upgrades:** Automatically get all updates and upgrades for the components in ComponentOne Studio for one year from your date of purchase.
- **New ActiveX Releases:** Get any new ActiveX components that we release during the term of your subscription.
- **Free e-mail support:** Receive e-mail support for one year from your date of purchase.

Visit www.componentone.com/studio to download your eval today!

\$399.95 Upgrade Price

\$499.95 Full Price

ComponentOne™

www.componentone.com

1.800.858.2739 or 1.888.228.4839

© 2001-2002 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.



Visual Studio Magazine™

Now Incorporating

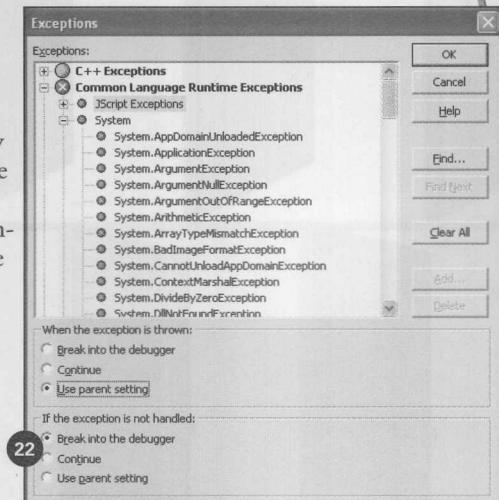
Visual Basic Programmer's Journal • Visual C++ Developers Journal • SQL Pro

F E A T U R E S

22**Debug VS.NET Applications**

VS.NET's debugging tools can help you solve a variety of debugging problems. See how to use these tools, along with various problem-solving techniques, to solve even the thorniest debugging issues.

by Nancy Folsom and Kathleen Dollard



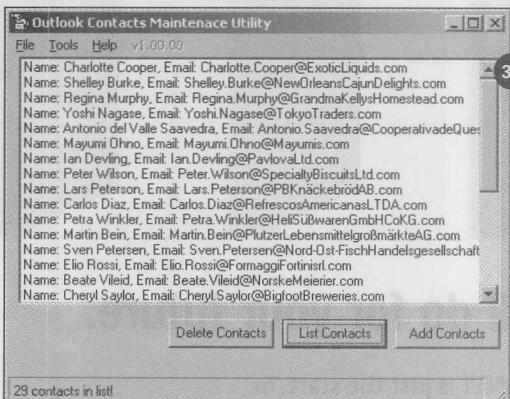
22

C O L U M N S

34**Getting Started****Manage Contacts With Outlook and VB**

Access Outlook's object model from VB6 to create and maintain contacts lists.

by Stan Schultes



34

www.visualstudiomagazine.com
To subscribe:**866-387-8275 (domestic),****850-682-7652 (international),****or 800-456-5911**

Visual Studio Magazine

| Title | Author | Keywords |
|---|---|--|
| July 2002 | | |
| Build a Scalable Architecture Create a Trickle-Feed Application Getting Started: Create and Consume XML Web Services Desktop Developer: Evaluate Strings and Convert Data in .NET Q&A: Drag and Drop URLs | Kathleen Dollard Zane Thomas Stan Schultes Edward G. Nilges Karl E. Peterson and Javal Lowy Ken Cox | VB.NET, VB6, ADO.NET, n-tier C#, .NET assemblies, application domains VB.NET, ASP.NET, XML, VB6, Web services, WinForms, SOAP, Web Forms VB.NET, strings, data conversion |
| ASP.NET: Implement Client Scripts Database Design: Compare XML Data Access APIs Black Belt: Exploit .NET Objects Create a Data Upload Service Web Services Introspect to Improve Security Detect Windows Version Directly Editor's Note: Visit Our New Home Guest Opinion: .NET Migration Works—Kind Of | Gray Ewing and Jon Rauschenberger Alberto Falossi Matt Hostetler Lex The Hank Marquis Patrick Meader Bill Wagner | C#, VB.NET, client-side scripts C#, SQL Server 2000, ASP.NET, VB6, XML, data access VB.NET, C#, .NET objects ASP, ADO, XML, XSLT, data upload, SQL Server 2000, VB6, SOAP Toolkit Web Services Interoperability Organization, Neil Charney, Chris Kun VB6, VB.NET, Windows OS, Windows version detection Fawcette Technical Publications, online catalog, www.visualstudiomagazine.com .NET migration, .NET Framework, Managed C# |
| June 2002 | | |
| Salaries Stabilize in Tough Market Getting Started: Create Random Access Log File Desktop Developer: Craft .NET Controls With Inheritance Q&A: Develop Rich-UI Apps | Susannah Pfalzer Billy Hollis Michael Welch Karl E. Peterson and Javal Lowy Chris Kinsman | VB.NET, subclass controls, custom controls, owner-draw listbox salary survey, salaries, economy VB6, log files, random access, user-defined type custom controls C#, VB6, Windows Forms, MDI, MFC, ASP.NET, server controls |
| ASP.NET: Provide Rich Functionality With Server Controls Database Design: Create More Efficient Database Code Black Belt: Decouple Components With LCE Explore Binary Compatibility With .NET Metadata Build Plug-and-Play Web Portals Improve ASP.NET Application Performance Secure Your Web Services Editor's Note: Real-World .NET | Dino Esposito Javal Lowy Javal Lowy Lex The Paul Deloglino Wei Meng Lee Patrick Meader Bill Storage | SQL Server 2000, ADO.NET, .NET Data Provider, database code C#, loosely coupled events, LCE, decoupl C#, C++, metadata, binary compatibility Web Services for Remote Portals, WSRF plug and play, OASIS, Thomas Schack VB.NET, SQL Server 2000, VB6, ASP.NET IIS, IES, JavaScript, Web services, ecr VB.NET, ASP.NET, XML, SOAP, NET Framework, Web services, ecr Visual Studio .NET, VB.NET, C#, corporate applications VB.NET, productivity |
| May 2002 | | |
| Invoke Asynchronous Magic Access Windows Networking Info Enhance SQL Server Security Getting Started: Master .NET's New String Objects Desktop Developer: Detect Your Windows Version Q&A: Retrieve Command-Line Arguments | Robert Teixeira L.J. Johnson Walter Myers III and David Byers Stan Schultes Hank Marquis Karl E. Peterson and Marius Sjögren | VB.NET, background tasks, asynch VB6, networking, Windows 9x SQL Server 2000, VB6, ASP, ADO Windows 2000, VBScript, security VB.NET, string objects, .NET 1 StringBuilder VB.NET, VB6, Windows verif C#, VB6, command line, Get CD AutoRun, System.Environment |

61

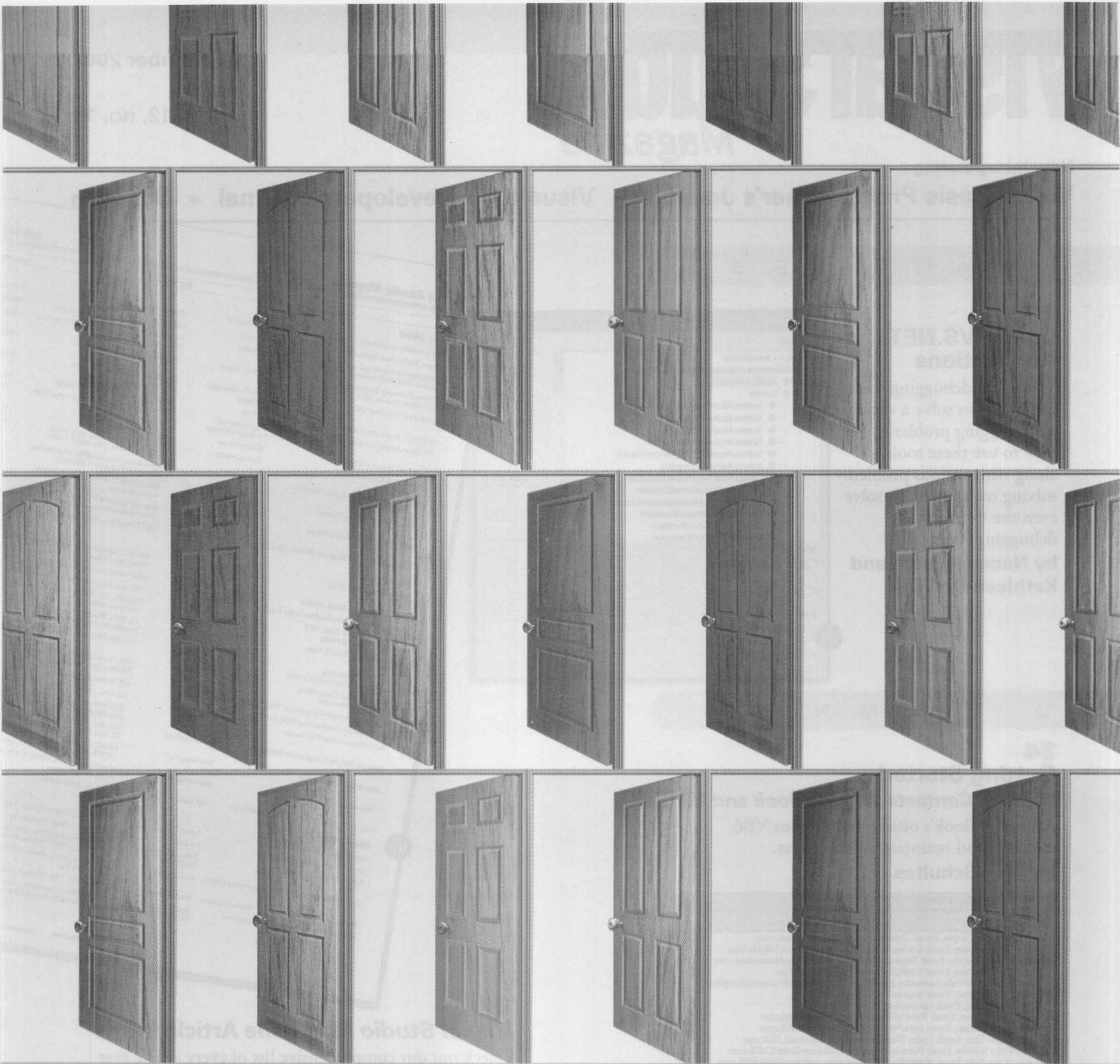
61

Visual Studio Magazine Article Index

Check out our comprehensive list of every article ever published in *VSM*. After you search the keywords for a topic you're interested in, you can read the related article for free on our Web site (www.visualstudiomagazine.com) by typing in the associated Locator+ code.

More contents ►

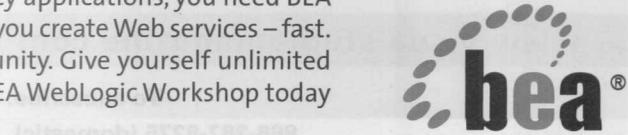
PRINTED IN USA



What BEA WebLogic Workshop will do for your future.

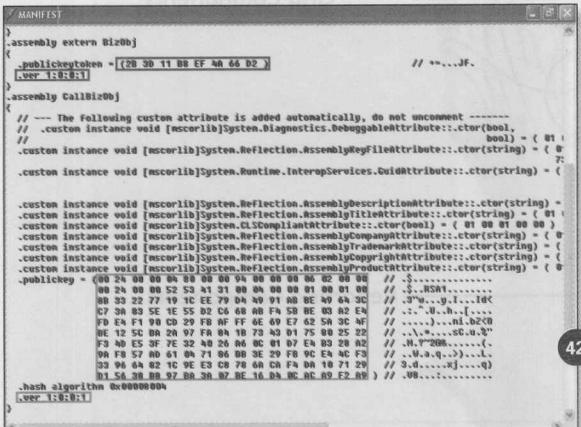
If you want your Web services to work across and beyond your enterprise, .NET is just the start. To interface with the overwhelming majority of J2EE, host and legacy applications, you need BEA WebLogic Workshop, the new development framework that helps you create Web services – fast. You'll also have all the support of our dev2dev developer community. Give yourself unlimited opportunities and open doors that .NET alone can't. Download BEA WebLogic Workshop today at <http://dev2dev.bea.com/tryworkshop>.

The fastest route to enterprise Web services.



BEA and WebLogic are registered trademarks, and BEA WebLogic Workshop is a trademark of BEA Systems, Inc. © 2002 BEA Systems, Inc.

COLUMNS



42

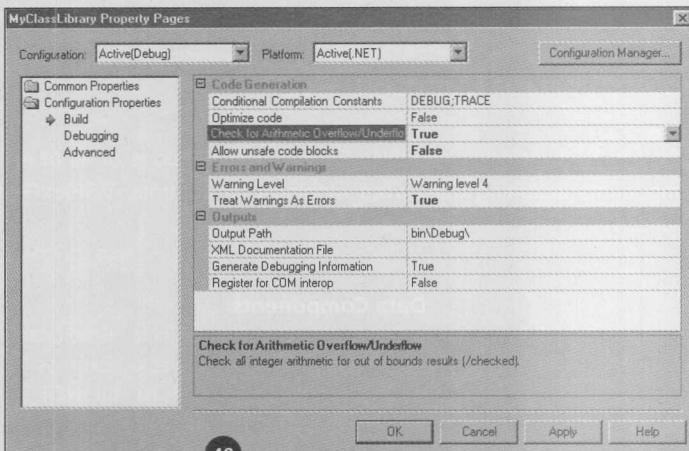
42

Desktop Developer

Deploy WinForms on the Web

Marry the deployment advantages of Web applications with the rich user experience and security of Windows applications.

by Mauro Sant'Anna



46

46

Q&A

Keep a Window Visible

Our Q&A columnists show you how to prohibit users from moving windows offscreen, use unsafe code the right way, work with unchecked code, and declare constant object variables.

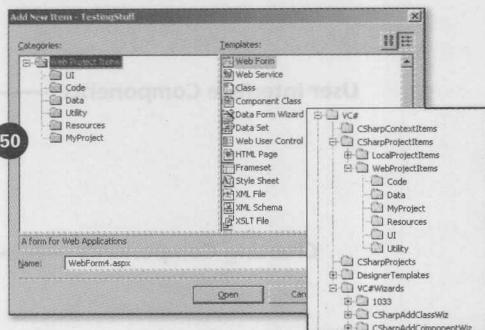
by Karl E. Peterson, Juval Löwy, and Mattias Sjögren

50 ASP.NET

Create Boilerplate Templates

Building a boilerplate template saves you time and repetition. Create your own project item template to customize the code VS.NET creates for Web Forms.

by Jonathan Goodyear



50

54

Database Design

Manipulate Schema Information

ADO.NET doesn't provide classes for schema manipulation, but you can modify schemas programmatically. Find out your three schema manipulation options and when to use each one.

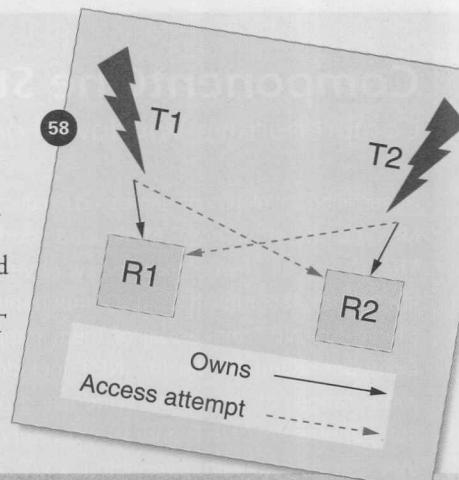
by Dino Esposito

58 Black Belt

Sync Threads Automatically

Use automatic thread synchronization to prevent deadlocks and simplify the management of .NET component concurrency.

by Juval Löwy



DEPARTMENTS

11 Editor's Note

by Patrick Meader

12 Letters to the Editor

15 First Looks

67 Index of Advertisers

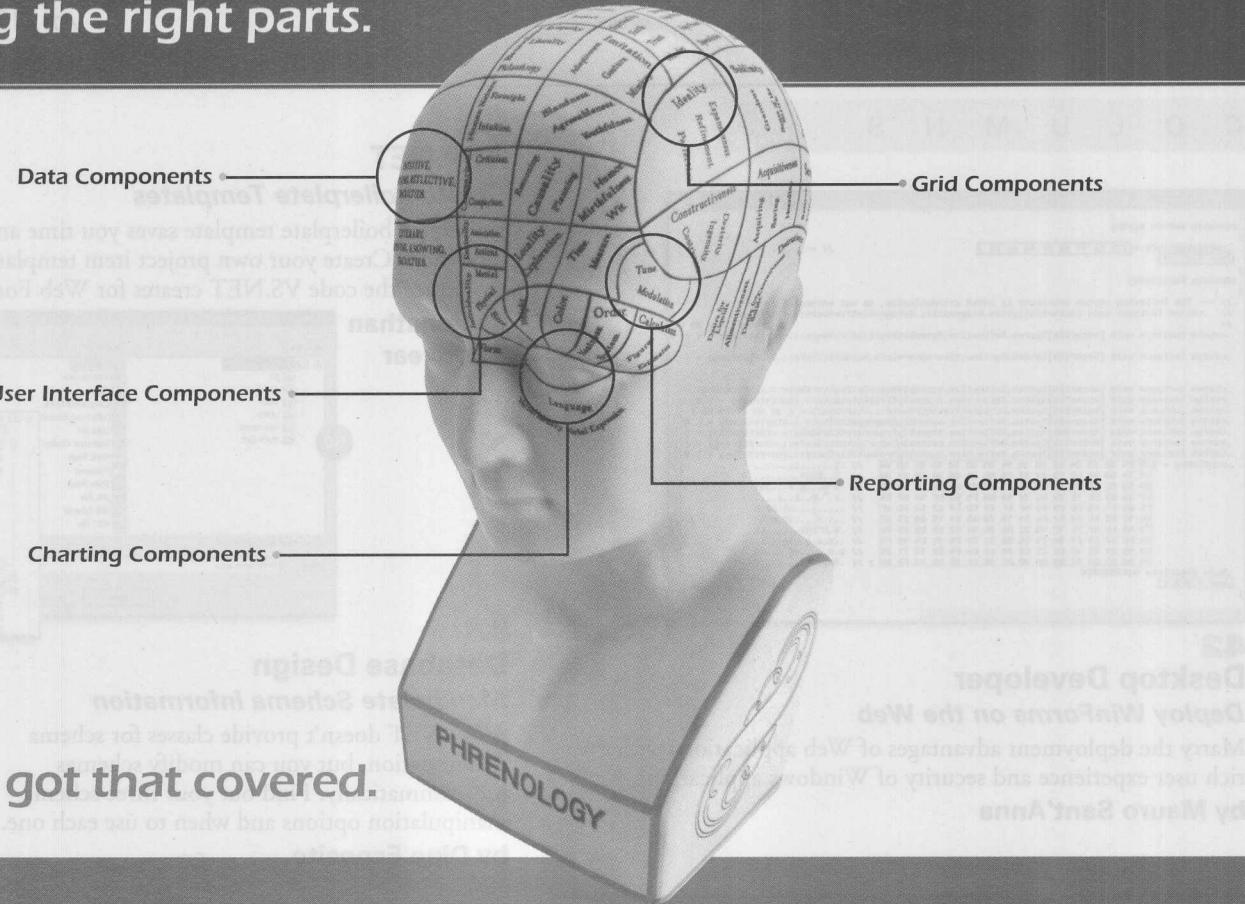
72 Guest Opinion

by Nancy Folsom

Online contents ▶

Customer Service: For subscription orders, inquiries, or address changes, call toll-free 866-387-8275 (for domestic subscribers), 850-682-7562 (for international subscribers), or 800-456-5911; send a fax to 650-321-3818; send an e-mail to vsmcustomerservice@neodata.com; or write to *Visual Studio Magazine*, P.O. Box 58872, Boulder, CO 80322-8872. Foreign and Canadian orders must be payable in U.S. dollars plus postage. The surface rate to Canada and Mexico is \$52.97 per year and for all other countries, the air mail rate is \$78.97 per year. **Back Issues:** To order *Visual Studio Magazine* back issues, call 650-833-7100 or 800-848-5523 and ask for the customer service department. Back issues cost \$10. Additional postage will be charged to deliveries outside the USA. **Reprints:** For all *Visual Studio Magazine* editorial and advertising reprints, please contact PARS International Corporation at 102 West 38th Street, 6th Floor, New York, NY 10018; call 212-221-9595; or fax 212-221-9195. **Permissions:** If you would like to quote from an article, please fax your request to Tami Stoeckle at 650-853-0230, or mail the request to 913 Emerson St., Palo Alto, CA 94301-2415, Attn: Tami Stoeckle. Specify the issue date, title of the article, the portion you would like to quote, and the purpose. **Photocopy Rights:** Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Tami Stoeckle for more information. © Fawcette Technical Publications, all rights reserved. "Visual Studio" is a trademark of Microsoft Corporation and is used by Fawcette Technical Publications under license from the owner. *Visual Studio Magazine* is an independent publication not affiliated with Microsoft Corporation. Microsoft Corporation is not responsible in any way for the editorial policy or other contents of the publication. All contents of *Visual Studio Magazine* are copyright © 2001 by Fawcette Technical Publications, unless otherwise noted. "VBITS," "Interactive Developer," "Java Pro," and "Inquiry.com" are trademarks of Fawcette Technical Publications, a California Corporation, James E. Fawcette, President. "Java" is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication.

The first rule of intelligent Microsoft .NET development is having the right parts.



We've got that covered.

ComponentOne Studio™ for .NET

Comprehensive Windows Forms Components Subscription for Microsoft® .NET

Streamlined development. Increased Productivity. Maximized efficiency. Minimized coding. These are all things that every developer strives to accomplish and ComponentOne Studio for .NET makes all of them happen. ComponentOne Studio for .NET includes a wide range of powerful Windows Forms components for .NET and support services that equip you with everything you'll ever need to develop cutting-edge applications for Microsoft .NET.

ComponentOne Studio for .NET is a comprehensive components subscription that provides unsurpassed business value by delivering the latest Windows Forms components for .NET, updates and upgrades, new software offerings, and e-mail and online support for one full year from your date of purchase. ComponentOne Studio for .NET is the intelligent solution for .NET development.

ComponentOne Studio for .NET includes Windows Forms .NET versions of all of our most popular ActiveX® components including our award-winning grid controls, VSFlexGrid® Pro and True DBGrid® Pro.

Grid Components

- ComponentOne FlexGrid® for .NET
- ComponentOne True DBGrid® for .NET

Reporting Components

- ComponentOne Preview™ for .NET
- ComponentOne Reports™ for .NET

Charting Components

- ComponentOne Chart™ for .NET

Data Components

- ComponentOne DataObjects™ Enterprise for .NET
- ComponentOne DataObjects™ Express for .NET
- ComponentOne True DBList™ for .NET
- ComponentOne Zip™ for .NET

User Interface Components

- ComponentOne Spell™ for .NET
- ComponentOne Menus and Toolbars™ for .NET

Subscription Benefits

- **Free updates and upgrades:** Automatically get all updates and upgrades for the components in ComponentOne Studio for .NET for one year from your date of purchase.
- **New Windows Forms Releases:** Get any new Windows Forms components that we release during the term of your subscription.
- **Free e-mail support:** Receive e-mail support for one year from your date of purchase.

Visit www.componentone.com/studionet to download your eval today! \$399.95 Introductory Upgrade Price \$599.95 Introductory Full Price

ComponentOne™

www.componentone.com

1.800.858.2739 or 1.888.228.4839

© 2001-2002 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.



ONLINE

Visit *Visual Studio Magazine* online to read these exclusive articles and more:

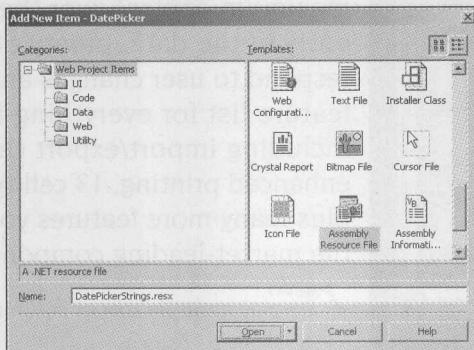
Create Localizable Web User Controls

Locator+ code:

VS0209GV_T

International-aware Web controls can reach out to foreign markets. Create a Date Picker Web control that supports different cultural settings.

by Gerardo Villeda

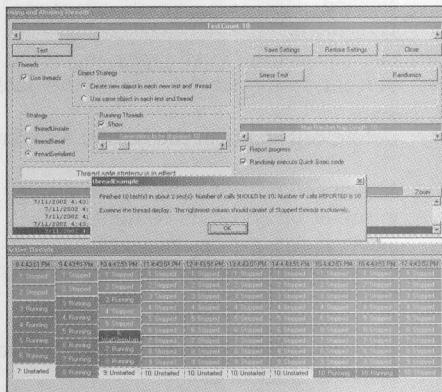


Increase Availability and Performance With Threads

Locator+ code: **VS0209EG_T**

Delve into this set of proven best practices for multithreading in VB.NET. Master these and you'll find your code running faster and cleaner.

by Edward G. Nilges



Oracle Provider for .NET Extends ADO.NET

Locator+ code:

VS0209LT_T

VSM interviews George Demarest, Oracle's director of database marketing, on what Oracle is doing for .NET developers—and how Microsoft could help.

by Lee Thé



HOT .NET TIPS

Locator+ code: **VS0209TT_T**

Check out these helpful .NET tips and tricks from VSM authors:

- **Change VS.NET Keyboard Mappings** by Dianne Siebold

- **Use the Marker Interface Pattern** by Mattias Sjögren

- **Make the Most of VS.NET's Start Page** by Juval Löwy

- **Page and Sort Resultsets With ASP.NET** by Mauro Sant'Anna

- **Optimize VB.NET Code Performance** by Francesco Balena

- **Tag Your Objects to Anticipate Change** by Edward G. Nilges

Locator+ Defined

Type in this Locator+ code to download the code for this particular article. You can also get the code for this article by pulling up the article itself, then clicking on the link to the accompanying code.

Type in this Locator+ code to discuss this article with the author and other readers in VSM's online discussion forums.

This Locator+ code brings up the article itself, along with any code listings, sidebars, and other extra content that wouldn't fit in the printed edition of the magazine.

This Locator+ code brings up a past article on a related topic.

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209GV Download the code for this article.

Discuss

VS0209GV_D Discuss this article in the ASP.NET forum.

Read More

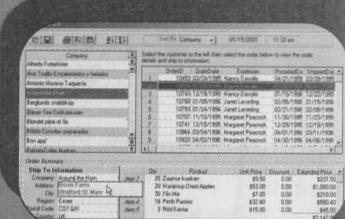
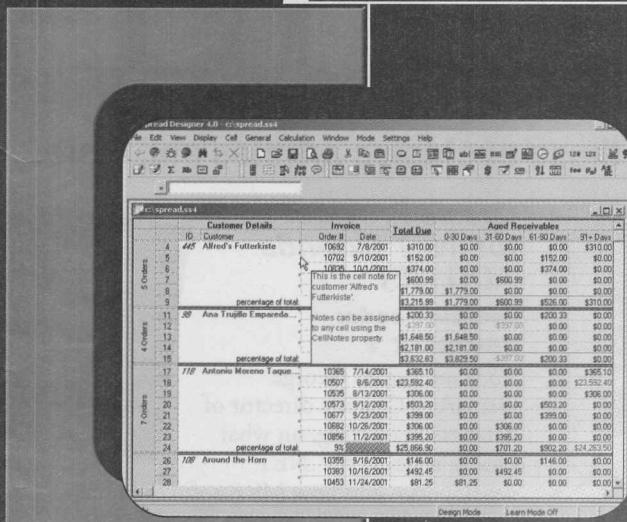
VS0209GV_T Read this article online.

VB9807GS_T Getting Started, "Store Language Strings in a Resource File," by Chris Barlow

.NET NEWSLETTER SIGN-UP

Every week, the *.NET Insight* e-mail newsletter brings you up-to-date news, technical information, opinions, interviews, and analysis on topics and technologies such as Exchange, Visual Basic .NET, C#, SQL Server and data access, ASP.NET, .NET Servers, wireless, Web Services, and XML. Sign up for free at www.visualstudiomagazine.com.

✓ Spread 6



Premier developer and supplier of
Web Form, Win Form and COM Components

www.farpoint.biz

Use FarPoint's Spread 6 to easily incorporate advanced grid and high-level spreadsheet features into your applications. You get unmatched flexibility at the cell level for maximum control over the display and entry of the data, plenty of events to respond to user changes and an impressive feature list for everything in between; including import/export capabilities, enhanced printing, 13 cell types, formulas, plus many more features you'd expect from the market-leading component.

- ✓ **NEW!** Display multiple row or column headers
- ✓ **NEW!** Merge cells that contain identical data for easier viewing
- ✓ **IMPROVED!** Use any of the 13 cell types, including the new percent, to format the entry of the data
- ✓ **NEW!** Define cell spans in the header and data rows
- ✓ Attach cell notes to any cell to display additional information about its contents
- ✓ **NEW!** Automatically change the color of negative numbers to red
- ✓ Use the border styles to highlight an individual cell or a block of cells
- ✓ **IMPROVED!** Use the Spread Designer to facilitate rapid development using its' point-and-click design-time interface
- ✓ Use the Print Preview control to see how your printed spreadsheet will look
- ✓ Users can sort by clicking column headers using built-in sort
- ✓ Bind the spreadsheet to ActiveX Data Objects (ADO) 2.0 using OLE DB technology, Data Access Objects (DAO) and Remote Data Objects (RDO)

FARPOINT
TECHNOLOGIES INC
www.farpoint.biz
1-800-645-5913

Goin' Mobile With MMIT

The Microsoft Mobile Internet Toolkit (MMIT) is to the impending Compact Framework what ASP/ASP.NET is to desktop Visual Studio development. That's a good thing for you as a developer. Microsoft actually groups MMIT under ASP.NET. Like ASP and ASP.NET, MMIT depends on server-based extensions interacting with IIS to kick down HTML to browsers on portable devices, whether Pocket PC devices, Palm OS devices, or Web-enabled mobile phones.

This has all the pluses and minuses you'd expect. The Compact Framework lets you create standalone applications that you install and run on portable devices such as Pocket PC, Pocket PC Phone, and the upcoming Smartphone, whereas MMIT requires that your device have an active connection to the Internet before you can use an MMIT-based application. It also goes without saying that you can create far richer applications using the Compact Framework on Pocket PC and related devices than you can using HTML through a browser—much as you can create richer standalone apps on Windows than going through a Web interface.

The biggest drawback to the Compact Framework is that it targets Pocket PC and related devices exclusively. You can't target Palm devices with it, and you can't target mobile phones that aren't Pocket PC-enabled. That's significant because Sun and its various Java partners have done a much better job getting Java into mobile phones and other mobile devices than Microsoft has to date. You can already buy Java-enabled phones from Nokia, Motorola, and many others. Meanwhile, Microsoft's Compact Framework-enabled Smartphone technology remains on the horizon, due later this year or early next, and the first iteration won't even ship with the .NET runtime for Pocket PC.

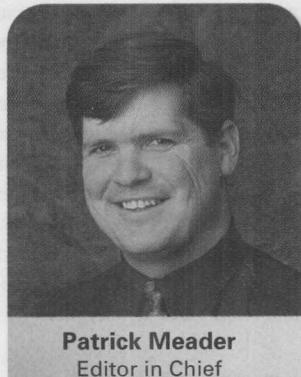
Ninety percent of winning is showing up, the adage goes, but Microsoft is late to the game in the arena of programmable mobile phones. The reasons aren't really important—it might be because mobile phone companies, like set-top box companies before them, avoided alliances with Microsoft because they feared being vulnerable to an embrace-and-extend strategy. The fact remains: Mobile-device manufacturers have been slow to embrace Microsoft-authored programming languages in mobile phones, whether it's .NET, Visual Basic .NET, or something else.

On the positive side, Microsoft appears to be doing better in the area of PDA-style mobile devices after getting off to a slow start. Its Pocket PC technology continues to mature and improve at a terrific clip, both in terms of its basic functionality and in the

way Microsoft has streamlined its interface for the end user. Better still, VB.NET developers will have a much more comfortable experience under the Compact Framework than when using pre-.NET versions of VB to program mobile devices.

Microsoft's recent Pocket PC advances notwithstanding, corporate environments still face a mix of wireless Pocket PC and Palm-based PDAs for the foreseeable future. Unless a company standardizes on one or the other, it cannot create client-side applications for both sets of devices with a single code base. And no company wants to create the same application twice. Some companies do standardize on one or the other, but as often as not, employees buy the devices they prefer on an ad-hoc basis, leading to a highly heterogeneous mobile PDA environment. This is where MMIT shines. MMIT lets you target a wide range of portable devices with a single set of source. You can target a range of computer desktops and devices from a single application. A computer has different capabilities than a PDA, which has different capabilities than a mobile phone, which has different capabilities than a BlackBerry or other wireless pager. All these devices have different form factors, and it doesn't make sense to present the same UI to all the users of these various devices. MMIT lets you create a single form/application that targets all these devices, and you can have the IDE warn you when a given feature exceeds the capabilities of a device you're targeting, letting you make adjustments as necessary.

And this, for me, is why MMIT is what .NET is all about. It lets you target the users you need to target—whatever devices they might be using, wherever they might be using them—with out having to standardize on a particular mobile device. Yes, you trade off some functionality, but the world you can target expands tremendously. It's a chance to realize the promise of the Web all over again. **VSM**



Patrick Meader
Editor in Chief

What do you think of the Microsoft Mobile Internet Toolkit or Microsoft's mobile strategy in general? Discuss this with me in the *Talk.Editors* forum on our Web site. Use this Locator+ code: **VS0209EN_D**

Letters

Letters to *Visual Studio Magazine* are welcome. Letters must include your name, address, and daytime phone number to be considered for publication. Letters might be edited for form, fit, and style. Please send them to Letters to the Editor, c/o *Visual Studio Magazine*, 913 Emerson St., Palo Alto, CA 94301; fax them to 650-853-0230; or e-mail them to vsmedit@fawcette.com.

VB.NET: Too Far, or Not Far Enough?

So Bill Storage doesn't think VB.NET goes far enough [Guest Opinion, "VB.NET = .NET ENOUGH," VSM June 2002]. It's not a lot of fun converting real-world apps to VB.NET. The apps I'm converting aren't spaghetti-code monstrosities or VB3 throwbacks—they're VB6 apps that were subjected to exhaustive coding standards based on coherent object-oriented principles.

Bill, what's the attraction of zero-based collections? I'm glad VB.NET has the same one-based collections I've used everywhere for years. Why would I want to screw everyone who uses my objects by making my

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS209: All the listings and code files for the September 2002 issue of VSM in one ZIP file
VS209NF: "Debug VS.NET Applications": two sample projects illustrating different types of bugs you might encounter in your application

VS209GS: Getting Started: the ContactsUtility sample application that uses Outlook's object model to maintain Outlook contacts from an Access database table; plus two extra code listings that show you how to add and update Outlook contacts

VS209DT: Desktop Developer: the business DLL, two apps that call it, an app that uses Windows controls, and a DLL with Windows Forms inside

VS209QA: Q&A: VB6 code demonstrating how to keep a form fully onscreen, and VB.NET and C# code showing how to use read-only variables

VS209DD: Database Design: a VB.NET project that uses the DDL functions described in this article to create a new SQL Server database programmatically and query it for schema information

VS209BB: Black Belt: a sample component that uses a synchronization domain and a test client, both provided in C# and VB.NET

Read More

VS209LE_T: Read Letters to the Editor online. They include extra letters not printed due to space constraints.

collections zero-based when they've previously been one-based? Zero-based collections are a throwback to another time. My clients count from one, and I count from one.

I wish VB.NET had used a little more of VB6's common sense. Was it necessary to remove the `UnloadMode` parameter from the form's `QueryUnload` (now `Closing`) event? VB.NET help says you need to write your own logic to determine how the form is being closed, but what was wrong with forms knowing how they're being closed?

Is it necessary to force developers to use circuitous calls such as `System.Diagnostics.FileVersionInfo.GetVersionInfo(System.Reflection.Assembly.GetExecutingAssembly.Location).FileBuildPart` instead of VB6's elegant `App.Revision`? I realize that using Imports means I don't have to type all of it, but I still have to know all of it at some point to import the appropriate namespace. Doesn't the .NET world tolerate any of VB6's common-sense shortcuts?

Bill, the "don't break my code" argument is economically sound. If it takes months to convert an app to VB.NET, then those months can't be spent making any money for the companies I work for. Your hobby apps might be fun to overhaul completely, but my enterprise-critical apps are not.

Dave Doknjas, Surrey, British Columbia

I agree with Bill Storage, even though change isn't painless. The cost of doing things correctly is less than the cost of fixing things that aren't correct (or quirky).

Joe Shapiro, Kansas City, Mo.

I agree with Bill Storage. I think Microsoft did a severe injustice to the people who use VB who come from a C background. Certain things should just be when it comes to programming. If you rely on -1 to be true rather than use the cookie, then you deserve to be hit. If you don't like short-circuit if statements, then you should stay with VB6. I, for one, think that VS.NET beta 1, which followed the more standard approach to things, shouldn't have been compromised. At least give those who have a good programming background a way to have true zero-based arrays without the extra 1, short-circuit ifs, true as 1 or nonzero, and so on.

Also, I too feel that the loss of Edit and Continue, even if it would have been the C++ implementation, is devastating. This is an extremely handy and productive tool and when it was taken away, I almost didn't go over.

Troy B. Stauffer, Hilliard, Ohio

Visual Studio™ Magazine

GROUP PUBLISHER
Jeff Hadfield

VSM STAFF EDITORS

Patrick Meader, *Editor in Chief*
Nicole Pizzurro, *Managing Editor*
Lee Thé, *Executive Editor*
Susannah Pfalzer, *Executive Editor*

CONTRIBUTING EDITORS

Daniel Appleman, Francesco Balena, Chris Barlow
Andrew J. Brust, Matthew Curnel, Dino Esposito
Dan Fergus, Keith Franklin, Jonathan Goodyear
Mary V. Hooke, Roger Jennings, L.J. Johnson
Don Kiely, Chris Kinsman, Deborah Kurata, Juval Löwy
Bill McCarthy, Karl E. Peterson, Keith Pleas, Stan Schultes
Ron Schwarz, Yaser Shohoud, Dianne Siebold, Bill Wagner
Phil Weber, Jonathan Zuck

TECHNICAL REVIEWERS

Ash Rosaf, *Senior Technical Reviewer*
Karl E. Peterson, *Tech Tips Editor*
Jason Bock, Ken Cox, Don Dugdale, Alberto Falossi
Bill McCarthy, Ron Schwarz, Dianne Siebold

EDITORIAL ADVISORY BOARD

Francesco Balena, Richard Finkelstein, Craig Goren, Jim Lisiak
Bill Neville, Karl E. Peterson, Joe Robison, William Storage

ART & PRODUCTION

Michael Hollister, *Vice President, Art & Production*
Bruce Gardner, *Senior Art Director*
Casey Mallough, *Associate Art Director*
Brian Rogers, *Online Associate Art Director*
Michael Furukawa, *Associate Production Director*
Sheila Quirk, *Associate Production Manager*
Phil Weber, *Software Developer*
Richard Kwong, *Senior Producer*
Shane Lee, *Associate Web Producer*
James Do, *Associate Web Producer*
Yuliya Smirnova, *Web Design Intern*

ADVERTISING SALES

Karin Becker, *Group Advertising Director*
Rich Sims, *Regional Account Manager (Southwest)*
Mark Romasco, *Regional Account Manager (Northeast)*
Jeff Lorton, *Regional Account Manager (Northwest)*
David Seymour, *Inside Sales Manager*
Katriona Cepeda, *Inside Sales Representative*
Regan Moore, *Online Account Manager*

CIRCULATION

Karen Koenen, *Circulation Director*
Meg Clark, *Circulation Manager*
Joyce Hoelzle, *Circulation Manager*
Gerry Guzman, *Circulation Associate*

MARKETING

Libby Dickinson, *Marketing Communications Manager*
Margaret Horoszko, *Senior Marketing Designer*
Susan Ogren, *Marketing Specialist*
Helga Taylor, *Marketing Designer*
Christian Pfrommer, *Corporate Marketing Intern*

CONFERENCES

Tena Carter, *Vice President, Conferences*
Tim Smith, *General Manager, Conferences*
Tina Fontenot, *Exhibit Sales Manager*
Heidi Corcoran, *Conference Sales Planner*
Leslie Hasvold, *Conference Operations Planner*
Jennifer Brucker, *Marketing Manager*
Shannon Sullivan, *Marketing Planner*
Maryam Ghannaghani, *Associate Editor*
Lyndon Lloyd, *Associate Web Producer, Conferences*
Nicole Pickett, *Conference Marketing Intern*

FTPONLINE

Tim Haight, *Editor in Chief*
Nina Goldschlager, *Managing Editor and Business Unit Manager*
Pamela Olheiser, *FTPOnline Editorial Intern*

CUSTOMER SERVICE

Laura Groenewold, *Customer Service Representative*
Nicholas Cadigan, *Customer Service Representative*

OPERATIONS

Darlyn Phillips, *Director of Finance*
Neema Buruku, *Financial Analyst*
Elena Ostrovsky, *Accounts Payable Accountant*
Iain Neillands, *Accounts Receivable Accountant*
Lina Yonemitsu, *Collections Analyst/Cash Management Consultant*
Betty Wu, *Staff Accountant*
Philip Ziman, *HR and Facilities Director*
pamela Davis, *HR Manager*
Todd Gallagher, *Information Systems Specialist*
Tin Cao, *System Administrator*

FAWCETTE TECHNICAL PUBLICATIONS

James E. Fawcette, *President*
Patricia Cameron-Loyd, *Chief Operations Officer*
John Sutton, *Executive Vice President and Chief Financial Officer*
Matt Carter, *Vice President, FTPOnline*
Aaron Weule, *Vice President of Technology and Chief Information Officer*
Paul Spiksma, *Publisher, FTP Advertising Network*
Wilson, Sonsini, Goodrich & Rosati, *Corporate Counsel*



Crystal Decisions, Crystal Reports, Crystal Enterprise and Crystal Analysis, are either a registered trademark or trademark of Crystal Decisions, Inc. All other trademarks or registered trademarks referenced are the property of their respective owners.

But you need the right information first. Fox Filmed Entertainment knows this. That's how they've become one of the most successful creators and distributors of blockbuster films

in the world. And that's why they chose Crystal Decisions™. Our web-based reporting, analysis and information delivery technology helps Fox track critical business information through

WHAT KIND OF DECISIONS DO YOU MAKE IN HOLLYWOOD? IF YOU'RE FOX™, ENTERTAINING ONES.

distribution channels and leverage that information to make smarter decisions. With over 10 million licenses shipped, and partners including SAP® and Microsoft®, we've proven our

solutions deliver information, cut costs and improve productivity. Want to learn more? Visit: www.crystaldecisions.com/prad/002/, or call 1-866-821-3525.

Access. Analyze. Report. Share.™

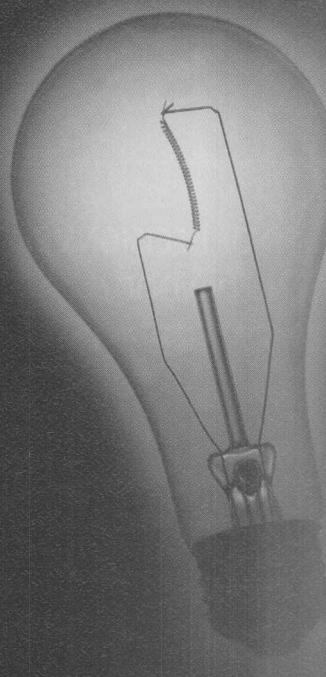


crystal decisions™

"I had the opportunity the other day of 'playing' with InstallShield®, and then it dawned on me what a great product I have in Wise for Windows Installer. It really does make installing our programs simple."

*-Martin Hart, Memory Soft S.L.**

© 2002 Wise Solutions Incorporated. All rights reserved. InstallShield is a registered trademark and Service mark of InstallShield Software Corporation. All other trademarks are the property of their respective owners.



"You-know-who" would have you believe that they are the only ones you should think of when creating software installations. Our Wise for Windows Installer tool has convinced thousands of developers otherwise. Wise was the first to offer an installation product for Microsoft's Windows Installer service, first to support the Microsoft .NET Framework, and the first to create 64-bit installations.

Take a stand for ease-of-use, reliability, and world-class technology and switch to Wise for Windows Installer

Want to know the other reasons why you should switch to Wise? Get the switch kit and find out at www.switchtowise.com or order today at 800-554-8565.

*Unsolicited testimonial from a satisfied Wise Solutions customer.
Full text of this quote can be found at www.wisesolutions.com



Software installations made easy™

Use UML With .NET

Rational XDE Professional v2002 is a serious software engineering tool for Visual Studio .NET. XDE, which stands for eXtended Development Environment, allows you to analyze, design, and document your .NET solutions fully within the VS.NET environment. XDE provides Unified Modeling Language (UML) support directly within .NET. You'll find this support particularly useful when you develop complex systems using C#.

XDE supports the standard UML diagrams by adding the Model Explorer and other modeling tools to VS.NET. You use the Model Explorer to work with class diagrams, use cases, and other UML models within your .NET solutions. You create these models using visual components that XDE adds to the .NET toolbox. Use XDE to model your solutions directly within the .NET environment, and

Rational XDE Professional v2002: .NET Edition

Rational Software

Web: www.rational.com

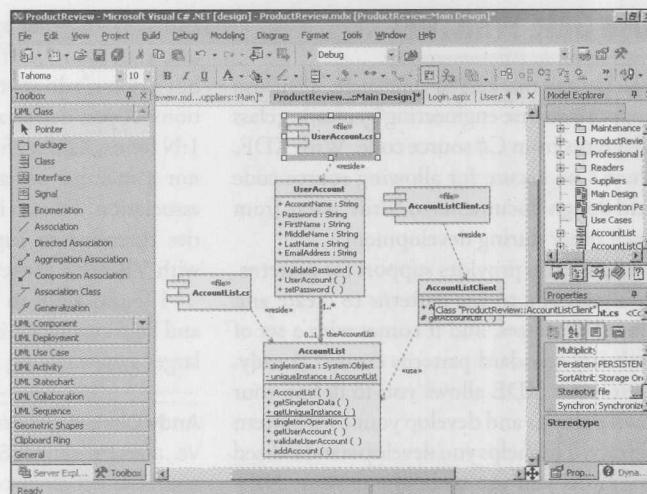
Phone: 800-728-1212; 408-863-9900

Price: \$3,595

Quick Facts: Adds UML and pattern templates to VS.NET.

Pros: Solid UML support; great C# integration; strong pattern support.

Cons: Steep learning curve if you don't know UML; expensive for smaller projects.



Integrate UML Visual Editing Into the VS.NET Environment. You can use Rational's eXtended Development Environment (XDE) to create UML diagrams with easy-to-use visual tools.

build these models using the same techniques you're using already to develop screens and other visual components. This is good .NET integration, but it gets better.

The C# integration is strong. The class diagram is the heart of UML software design. XDE supports seamless synchronization between class diagrams and C#. It generates and maintains C# classes as you change UML class diagrams, and it does this without

Manage ZIP Files

Inner Media's DynaZIP MAX/Active Delivery bundle is a comprehensive ZIP management toolkit. It supports a wide variety of programming contexts, including Visual Basic, the .NET languages, Borland's offerings, and several others. It addresses every imaginable ZIP management issue, including compression, directory recursion, multivolume creation and formatting, and the latest Deflate64 compression method. In addition to conventional archive management functionality, you can tie the library to Internet Information Server's (IIS's) Active Server Pages (ASP) interface for dynamic ZIP handling through Web sites you develop.

The self-extracting archive functionality is separate from the main ZIP management library package. The Active Delivery half of the bundle is available separately, as is the DynaZIP MAX package,

DynaZip MAX 5.0 and Active Delivery 3.0 Bundle

Inner Media

Web: www.innermedia.com

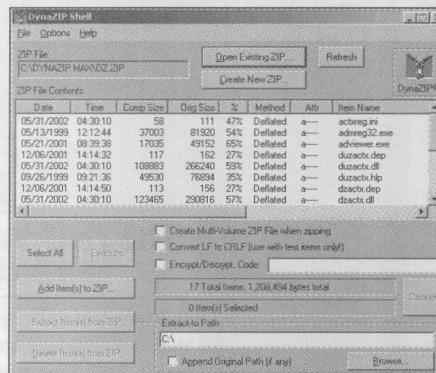
Phone: 800-962-2949; 603-465-3216

Price: \$384

Quick Facts: A comprehensive ZIP management package for developers and Web content providers.

Pros: Covers every imaginable ZIP/UNZIP scenario, including online ASP dynamic file handling.

Cons: Learning curve; lack of printed documentation (two large PDF manuals provided, total of 568 pages to read either on your monitor or in print).



Access a Variety of Wizards and Utilities. The DynaZIP Shell is an interactive front end for creating and managing ZIP files. Other utilities help you test and analyze archives and create scripts for self-extracting executables. The actual library code works through access to its API sets, accompanied by example code for a variety of programming platforms.

but there's a hefty discount of \$164 when you buy the bundle.

I was surprised by the depth of control and muscle I found in Active Delivery. I expected a typical "ZIP-to-EXE" wrapper. The package can do simple EXE packaging, but it can also create interactive installer packages from your ZIP archives. You can manage the engine's API from your own code if you need to create executable distributions dynamically. The engine requires proprietary script files, but it includes utilities to create and manage them, and there are API calls to handle the scripts through your code. The

Use UML With .NET

placing any markers in your code. XDE also supports reverse engineering by creating class diagrams from C# source code. With XDE, there's no excuse for allowing source code and design documents to "drift" away from each other during development.

XDE also provides support for patterns. It allows you to use patterns to create and populate classes, and it comes with a set of industry-standard patterns created already. Better yet, XDE allows you to define your own patterns and develop your own pattern library. This helps you develop standardized solutions you use throughout your .NET projects.

However, if you aren't familiar already with UML, then you'll need to learn it. XDE's documentation is good, but be prepared to make a significant investment to become UML-savvy.

I found several things that could be improved within the XDE product. The help is decent, but it isn't dynamic help. The C#

and class diagram synchronization brings over class names, attributes, and operations, but it doesn't do anything with class associations. If you define an Order class that has 1:N Items, XDE defines neither a collection nor a member variable to implement the association. I'd also like to see the capabilities extended to support synchronization with VB.NET as well as C#.

I found XDE to be an impressive tool, and I'd recommend it to anyone developing larger systems using .NET. **VSM**

Andy Clark is a consultant in the Richmond, Va., area. He's an MCSD with extensive Oracle and SQL Server experience. Reach him at theandyclark@hotmail.com.

Manage ZIP Files

feature set dovetails nicely with DynaZip's ZIP creation/management functions.

Using this kind of power isn't a trivial endeavor. There's an abundance of documentation (all online in the form of PDF

and help files) and a wealth of example code for numerous platforms, but you still must confront a bit of a learning curve, especially if you haven't worked with ZIP archives before from a developer standpoint. Although you can perform most functions with a few lines of code, it's critical that you understand the underlying rationale for what goes into those terse snippets.

Those hurdles shouldn't dissuade you from examining this package, because they're endemic to the context rather than the product. The example code is well-documented and readable, and the online documentation is exhaustive. I like this package, and recommend it without hesitation. It's good value for the money. If you only use a fraction of its power, you're still way ahead of the return on investment (ROI) curve. **VSM**

Ron Schwarz lurks in rural Michigan. When he's not digging out of the mountain of e-mail in his inbox, he maintains his sanity by restoring classic cameras. He welcomes your comments at www.clubvb.com.



- Windows 2000 Server
- SQL Server 2000
- MS Access 2002
- ASP / XML / WAP
- ASP Components
- Live Statistics
- Active Directory
- Windows Media
- HostMechanic™
- Visual Studio.NET
- MS FrontPage 2002
- Microsoft® Office XP

**4 months
hosting
FREE!**

Visit innerhost.com/4for4.asp

"Have you .NET, yet?"

Shared Web Solutions

Develop, test and deploy your Web Services and applications on the .NET Framework at an affordable price. Dozens of FREE components and valuable features are included with every iNTERHOST Shared Web Solution!

Dedicated VM Solutions

These solutions were designed from the ground up with one goal in mind; to be able to offer full dedicated server functionality at a much lower price. Dedicated VMs are enabled with all of the key components of .NET!

Dedicated Solutions

Robust and highly secure, your server is pre-configured with the .NET Framework. An iNTERHOST Dedicated Solution is for the developer who demands total control, unmatched performance and top notch security!

Studies show, it's not if you'll move to .NET but when! Don't get left behind. iNTERHOST has everything you need to make .NET work for you!



**Win \$300.00 in books
for Developers!**
You choose the books!

Visit: www.innerhost.com/winbk

Microsoft
GOLD CERTIFIED
.net

Toll Free: 888-751-5267
www.innerhost.com
Complex Hosting in a Global Environment™

iNTERHOST



Write faster code

Control revisions

Model

Deploy

Automate the mundane

Simplify

Write code faster

Get a life

be liberated

Being among the first Microsoft® .NET developers may be exciting. But wasting time is not. That's where we come in. We've spent six years working with Microsoft, and we've developed a new solution for .NET that lets you work completely inside the Visual Studio® .NET shell. Want everything from modeling to code analysis right at your fingertips? Done. Want your model automatically updated when you write code and vice versa? Done. Want to get rid of alt/tab and static void Main ()? Done. Want to try out the products that can make you the best .NET developer on the planet? Go to www.rational.com/offer/dotnet4. Done.

Rational®
the software development company

Rational and the Rational logo, among others, are trademarks or registered trademarks of Rational Software Corporation in the United States and/or in other countries. Microsoft and Visual Studio are trademarks or registered trademarks of Microsoft Corporation. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective companies. ALL RIGHTS RESERVED. ©2001 Rational Software Corporation.

Generate Apps From Your Data

Dataphor is a promising virtual database from Alphora that's billed as an "Automated Application Development" tool. This means it's a relational database environment that generates Windows and Web applications instantly from your data.

Dataphor does this by analyzing the database tables, rules, relationships, and constraints and converting them to a visual interface that includes data validation. Dataphor's master-detail forms are easy to use, and these extensible interfaces might be all you need.

Dataphor can connect to multiple databases seamlessly and invisibly to the user. You can cross-connect to SQL Server, Oracle, and DB2 and incorporate them easily. Dataphor has its own query processor, data dictionary, and storage/retrieval system, which it

Dataphor 1.0

Alphora

Web: www.alphora.com

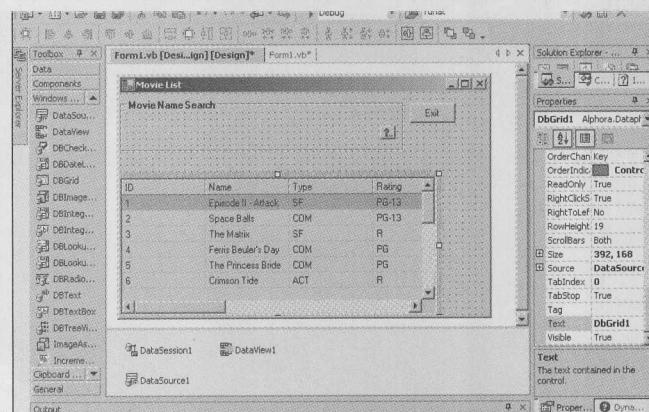
Phone: 801-371-6080

Price: \$5,000 single-developer license

Quick Facts: A database application system with added .NET components.

Pros: Connects to SQL Server, Oracle, and DB2; generates user interfaces for Web and Windows automatically.

Cons: Components can use only the Dataphor Data Application Engine.



View Data as You Program. Dataphor includes 25 data-aware .NET components that show data in real time as you develop.

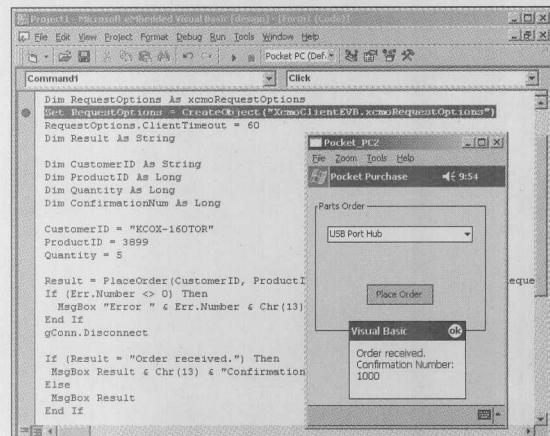
calls the Dataphor Data Access Engine (DAE). It also has its own query and management tool (Dataphoria) and its own data query language (D4). However, Dataphor is a "virtual" database and doesn't have its own database; instead, it manages other databases for improved performance, management, and capabilities.

Dataphor includes 25 .NET data-aware components that connect to the DAE. They range from the expected DBGrid to the snazzy Incremental Search. Dataphor also provides an ADO.NET data provider. These unique components show data as you program rather than at execution time. Dataphor accomplishes this by using a component architecture similar to what Borland has used in Delphi for several years. The architecture links to the DAE with a DataSession,

Connect PDAs to Server Objects

Extended Systems' XTNDConnect Mobile Objects (XCMO) are components that help you integrate personal digital assistants (PDAs) such as Pocket PCs and Palm handhelds with back-end servers and enterprise data. The SDK includes client, server, and monitoring software.

Exchanging data between a PDA and a database is fairly smooth with an all-Microsoft solution. However, companies with legacy databases, business objects, and Palm devices need to leverage their existing investments. XTNDConnect Mobile Objects provides a workable path to making COM objects and ActiveX Data Objects (ADO) or ODBC data available to small devices. What's more, you can build these solutions with AppForge,



XTNDConnect Mobile Objects Provide Connection Tools. You can connect a Pocket PC application easily to a back-end ODBC database by incorporating the XCMO components into your project.

CodeWarrior, or Embedded Visual Studio.

XCMO helps build the client- and server-side plumbing and wrappers that allow handheld computers to communicate securely with back ends. These COM-based solutions run over a wireless TCP/IP connection, a LAN, a WAN, or an Ethernet cradle.

For this review, I incorporated XCMO into a Pocket PC application using a VB6 server-side COM object and eMbedded Visual Basic (eVB) for the client. Setting up XCMO is fairly easy. One step installs the XCMO monitor service on the server. This directs traffic between the client and server on port 2032 (the

XTNDConnect Mobile Objects 1.2

Extended Systems

Web: www.extendedsystems.com

Phone: 800-235-7576; 208-322-7800

Price: \$395 per SDK, \$595 per server

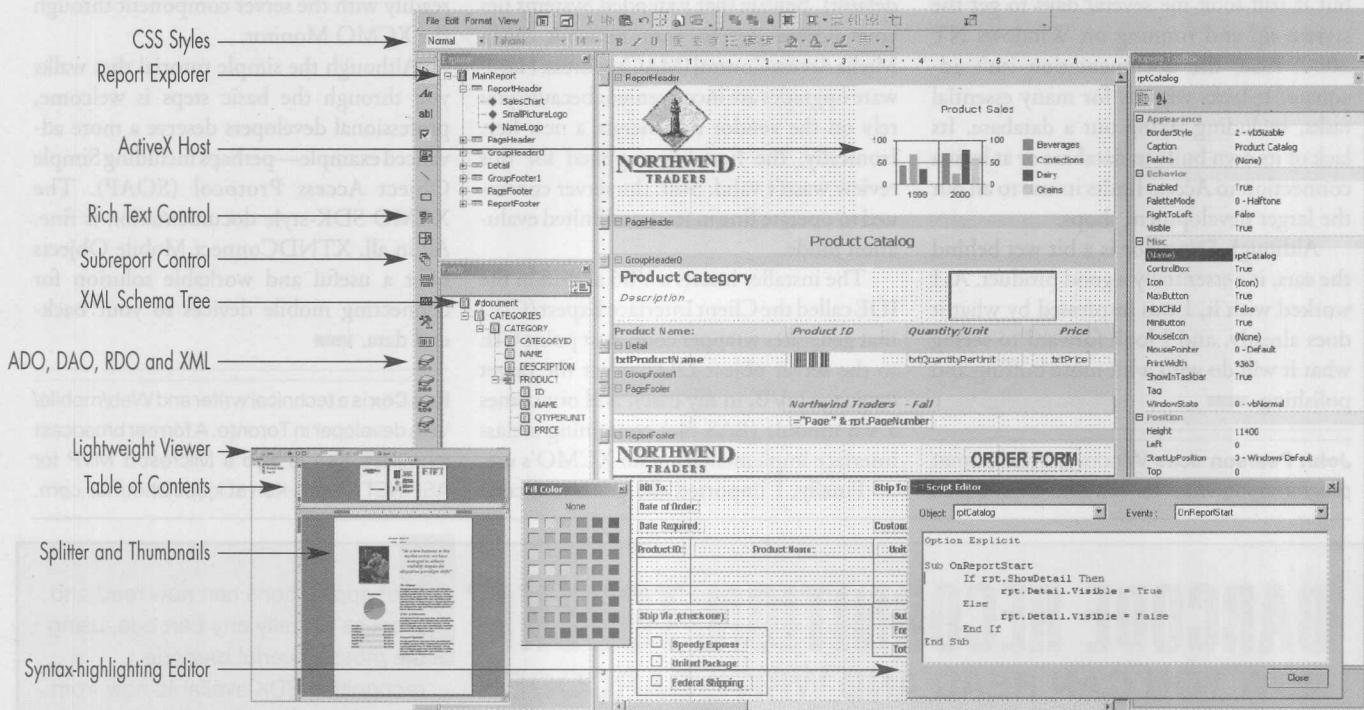
Quick Facts: Client-side components, development tool, and server components for integrating mobile devices with back-end systems.

Pros: Supports multiple platforms and programming languages; works well.

Cons: Example project too simplistic; license key tied to one network card.

Active Reports 2.0 is an upgrade to our best-selling reporting component. Data Dynamics offers **Active Reports 2.0** in two editions to suit your development needs. Both editions build on the power and ease of ActiveReports 1.0 and add many new features and enhancements based on your feedback.

Active Reports 2.0 Standard Edition now supports distributable XML report layouts, Active Scripting (VBScript/JScript) events and expressions and XML data sources. **Active Reports 2.0 Professional Edition** adds a royalty-free customizable end-user report designer control. Now your users can experience the power of ActiveReports within your applications.



Data Access

- Supports DAO, RDO, ADO.
- New! Support for XML data sources and XPath, XSL expressions.
- Supports unbound data from arrays, collections and other data providers.
- Binds to recordsets and resultsets for maximum flexibility and connection efficiency.

Architecture

- Full runtime access to objects, data sources and binding, ability to add controls at runtime to modify the report behavior.
- Full control over printer jobs, properties and output devices.
- Hosts ActiveX components and OLE Objects.

Performance

- Multi-threaded scalable engine.
- Fast response time.
- PRO! Report Caching service for fast web performance.

Design Environment

- Easy to use Office-like designer environment.
- New! Access Reports import wizard.
- New! Crystal Reports import wizard.
- New! VBScript and JScript events and expressions and a built-in syntax-highlighting script editor.
- New! ActiveReports 1.0 upsizer.
- Includes data bound text, checkbox, image, OLE objects, rich text and subreports.
- ActiveX controls host. Use 3rd party controls or your own custom controls in your reports.
- New! Barcode control.
- New! Report bookmark and internet hyperlinks.
- New! Controls Style property editor based on standard Cascaded StyleSheets (CSS) syntax.
- PRO! Royalty-free end-user report designer control.



Deployment

- Reports can be compiled for maximum security and speed.
- New! Report layouts can be distributed as XML-based report layout (RPX) files (application-host independent).
- Enhanced! Java viewer based on JDK 1.3 with printing support.
- New! TIFF export in addition to Text, RTF, Excel, PDF and HTML export filters.
- Enhanced! PDF export supports table of contents bookmarks and embedded fonts.

ActiveX Viewer

- New! Multipage thumbnails.
- New! Split views.
- New! Ruler.
- New! Text search.

Pricing

- Standard Edition: \$499
- Professional Edition: \$1299
(includes Royalty-free Report Designer)

Generate Apps From Your Data

creates a DataView, and finishes with a DataSource the components connect to. Unfortunately, these components connect only to the DAE and not to other data sources.

Dataphor is a complex product. I had complete access to the Alphora support staff, but it still took me several days to get the system up and running on Windows NT 2000. Also, the documentation isn't adequate. It lacks wizards for many essential tasks, including laying out a database. Its lack of its own built-in database or at least a connection to Access limits its use to all but the larger development shops.

Although Dataphor is a bit wet behind the ears, it's essentially a good product. As I worked with it, I was impressed by what it does already, and I look forward to seeing what it will do with a bit more buffing and polishing. **VSM**

John Pearson does Windows and Internet programming work for the Church of Jesus

Christ of Latter-day Saints. He has contributed to several magazines, including *New Architect* and *Java Pro*. You can reach John at compu_help@compuserve.com.

Connect PDAs ...

default). Beware that Extended Systems ties your license key to your network card's Media Access Control (MAC) address. Hardware upgrades are inconvenient because you rely on the vendor to generate a new key. Ironically, the first key supplied for this review wasn't valid. Still, the server continued to operate fine in its time-limited evaluation mode.

The installer inserts a VB6 add-in in the IDE called the Client Interface Expert (CIE) that generates wrapper code. Just point CIE to the server object DLL, select the target language (eVB, in my case), and out comes a VB module (BAS file) containing a class interface implementation for XCMO's use.

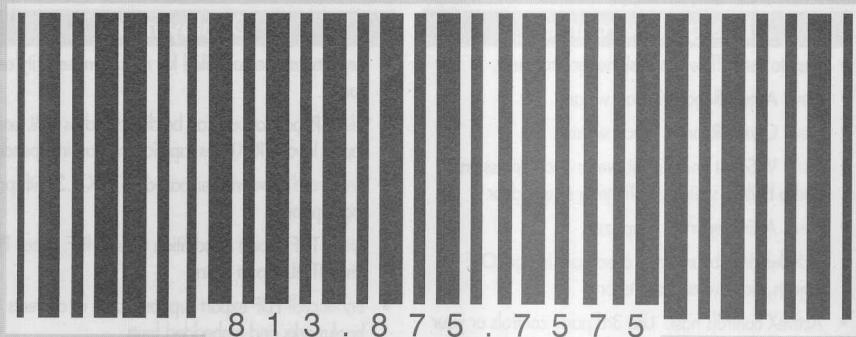
Finally, I imported the generated code

module into an eVB project and added a reference to the XCMOClientEVB type library. XCMO exposes functionality such as the xcmoInit method, which initializes the XCMO client (the Pocket PC) with server connection details. Within a couple hours, my iPAQ Pocket PC, in its cradle and using ActiveSync, was exchanging data readily with the server component through the XCMO Monitor.

Although the simple tutorial that walks you through the basic steps is welcome, professional developers deserve a more advanced example—perhaps including Simple Object Access Protocol (SOAP). The XCMO SDK-style documentation is fine. All in all, XTNDConnect Mobile Objects offer a useful and workable solution for connecting mobile devices to your backend data. **VSM**

Ken Cox is a technical writer and Web/mobile/VBA developer in Toronto. A former broadcast journalist, Ken is also a Microsoft MVP for ASP.NET. Reach Ken at kjopc@hotmail.com.

BARCODE RECOGNITION TOOLKIT NOW YOU CAN READ AND WRITE BARCODES!



DOWNLOAD SMARTSCAN XPRESS AND OUR
FULL RANGE OF DOCUMENT IMAGING
EVALUATION KITS AT:
www.pegasustools.com



PEGASUS IMAGING
CORPORATION



Your applications can now read and process virtually any barcode, using the most powerful barcode recognition SDK available now from Pegasus. SmartScan Xpress Barcode includes ActiveX controls for VB, class libraries for C++ and VCLs for Delphi.

SmartScan Xpress offers high-speed accurate readings of more than 20 industry 1D and 2D barcodes such as Code39, CODABAR, interleaved 2 of 5, Code 128, UCC128, EAN128, Code93, UPC-A, PDF417 and many more! It also offers the ability to write 1D barcodes.

SmartScan Xpress provides developers with a well-documented, easy-to-use interface, making project implementation a breeze!

(813) 875-7575 or (800) 875-7009
sales@pegasustools.com

TX Text Control Does it All



New! Now shipping v.8

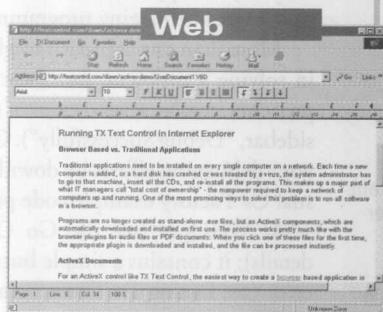
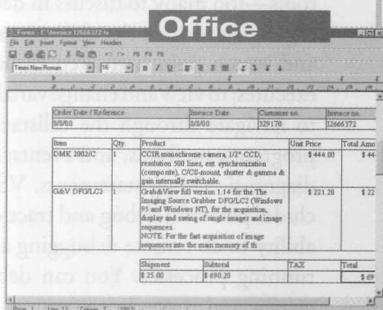
- Bullets & Numbered Lists
- Signed CAB File for Internet Download
- Print Method for Printing from IE, Access & FoxPro
- New Table Functions
- Extended C++ Class Library

Virtually all applications can make use of powerful word processing functions. TX Text Control gives you that power and delivers it in a reusable component form. With TX Text Control, you get hands-on access to the most comprehensive set of word processing features available in a development toolkit.

Database Applications

Text Control can be connected to any type of database, and store formatted text as binary data, RTF or HTML. You can store a complete document as a record, or load text from a database into table cells and macro fields. Typical applications which can be created are mail merge, report generators, formatted data entry masks, and all types of business apps. where documents are created from information stored in a database.

- Royalty-Free Word Processor Control
- ActiveX (Visual Basic, Delphi, VBScript,...) and Class Library (Visual C++)
- Read and Write RTF, HTML, Word 2000, ANSI and Unicode
- Run as .exe or in Internet Explorer (CAB file included)
- Tables, Headers & Footers, Hyperlinks, Bullets, Numbered Lists, Zoom, Page View
- FREE Tech Support



Text Control Features

- ActiveX for VB and Delphi
- Class Library for Visual C++
- File Formats: HTML, RTF, Word 6 to 2000, ANSI and Unicode
- Image Formats: TIFF, BMP, JPEG, GIF
- Tables
- Headers & Footers
- Bullets & Numbered Lists
- Text Flow Around Embedded Images, OLE Objects & Controls
- Undo-Redo
- Transparent Mode
- Read-only Mode
- Hyperlinks
- Macro Fields
- Connect to Any Type of Database
- Unlimited File Size (>5MB)
- Runs in Internet Explorer
- Drag & Drop
- Ready-to-Use Toolbars and Dialogs
- True WYSIWYG Page View
- Print Preview
- Zoom 10% - 400% in Steps of 1%
- Text Color and Background Color Adjustable per Character
- Localized for English, French, Spanish, Italian, German and Japanese. Can be Translated to Other Languages.
- Optional: Spell Checking with VideoSoft VSSpell
- Optional: Enhanced Image Processing with LEADTOOLS



877-898-2875

30 day money-back guarantee

To order: visit our website
or Email: sales@textcontrol.com

Web Applications

The included WebApps toolkit gives you all you need to deploy and run your application on the Internet:

- Use Tables and macro fields for calculations.
- Exchange .DOC files with MS Word
- Insert graphics or spreadsheets as OLE objects

The most powerful Text Control yet is available at:

www.textcontrol.com

VISIT FOR A FREE, TIME-UNLIMITED TRIAL VERSION

Debug VS.NET Applications

Solve debugging issues by using the VS.NET IDE debugging tools effectively.

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209NF Download the code for this article, which includes two sample projects illustrating different types of bugs you might encounter in your application.

Discuss

VS0209NF_D Discuss this article in the .NET Framework/IDE forum.

Read More

VS0209NF_T Read this article online.

N020319AN "Debug Apps Effectively With TraceContext" by Jonathan Goodyear

by Nancy Folsom and Kathleen Dollard

The best way to approach debugging depends on the problem, language, and your experience using the language. The best approach might change as you narrow down a problem's cause. In this article, we'll show you how to use the Visual Studio .NET IDE debugging tools to solve common problems.

VS.NET includes a wealth of debugging tools—too many to discuss in detail here. Some highlights include the ability to set breakpoints and watch points; to step through code as it executes; to view and change variables and values; to navigate through the callstack of executing programs, methods, and events; and to include diagnostic debug statements. VS.NET also includes powerful debug and trace classes, and the ability to do remote debugging and to attach to running processes. You can debug across languages and drop down into an assembly-level view of your application, and, to top it off, you can control debugging programmatically.

More than one approach might be successful in solving a problem, but most successful approaches involve several rules of thumb (see the sidebar, "Debug Successfully"). Get started with your debugging efforts by downloading the Visual C# Factory example code project from the VSM Web site (see the Go Online box for details); it contains a simple but common logic error (see Listing 1). The program should fill a shipment. When the code runs, an error dialog is displayed that says, "An unhandled exception of type 'System.IndexOutOfRangeException' occurred in Factory.exe. Additional information: Index was outside the bounds of the array." The error message doesn't say which array, or even what part of the code, is having the prob-

lem. So, you should elect to "Break." Once suspended, the IDE's debugging components help narrow down the source of the problem.

You can set breakpoints in the left margin (the red dot) of the IDE debugging Autos, Break-

Debug Successfully

Use these guidelines when solving debugging issues:

- Concentrate on one issue at a time.
- Make one change at a time as you experiment with solutions.
- Apply the scientific method to debugging efforts.
- Use source code control to log changes and to roll back failed experiments.
- Comment changed code with the date and reason for the change instead of deleting it.
- Use a consistent style, best practices, and common conventions to minimize bugs and to aid readability.
- Deviate from a best practice or common convention *only* after you know the rule you're breaking and why you're breaking it. Consider how you could rework the implementation you're considering to fit the convention.

points, and code windows (see Figure 1). Notice that ToolTips show the values of variables. Other debugging tools are available in tabs, such as Locals, Watch windows, Command window, and Breakpoints. In this example, the Autos window helps to identify the problem because the error refers to an array boundary, which involves variables' values. The Autos window shows you the counter (i) and the array (shipment).

Choose Between Locals and Autos

You can use the Locals window instead of Autos. The difference between the two is subtle. The Locals window is limited to showing the variables local to the currently executing statement. The Autos window is further limited to only those variables used on the current or adjacent lines.

Because i is equal to the number of crates in the shipment, and that's what you were expecting (according to the code), there might be something wrong with your loop counter. Expanding the "Shipment" array in the Autos window shows what the problem is: You've forgotten your array is zero-based. The solution is simple; you can change the For loop like this:

```
...
for (int i = 0; i < crates; i++)
...

```

Depending on your setting for "Changes in Visual Basic and Visual C# Code," found under Tools | Options | Debugging | Edit and Continue, it's possible to change the actual code in the For loop while debugging by selecting Edit | Replace to make the change. Once you continue the program, the IDE can either cancel the currently running program and restart with the code change, or continue running without incorporating the code change. Although this behavior is the default in C#, you must modify your options for this behavior in VB.NET. It's not precisely Edit and Continue from Visual Basic 6, but it's a useful feature.

Notice that the Breakpoints window in Figure 1 shows a condition:

```
when 'i==99'
```

Once you click in the margin in the code window to set a basic breakpoint, you can right-click on the breakpoint and choose Properties. Setting a condition on a breakpoint is a common need and is helpful when diagnosing problems that occur only on specific iterations, as in this case. If you break just before the error occurs, you can trace the conditions that set up the error. First, a word of warning: We inadvertently typed the condition as i=99, which changed the value of i while the program ran. So, the counter was always set to 99.

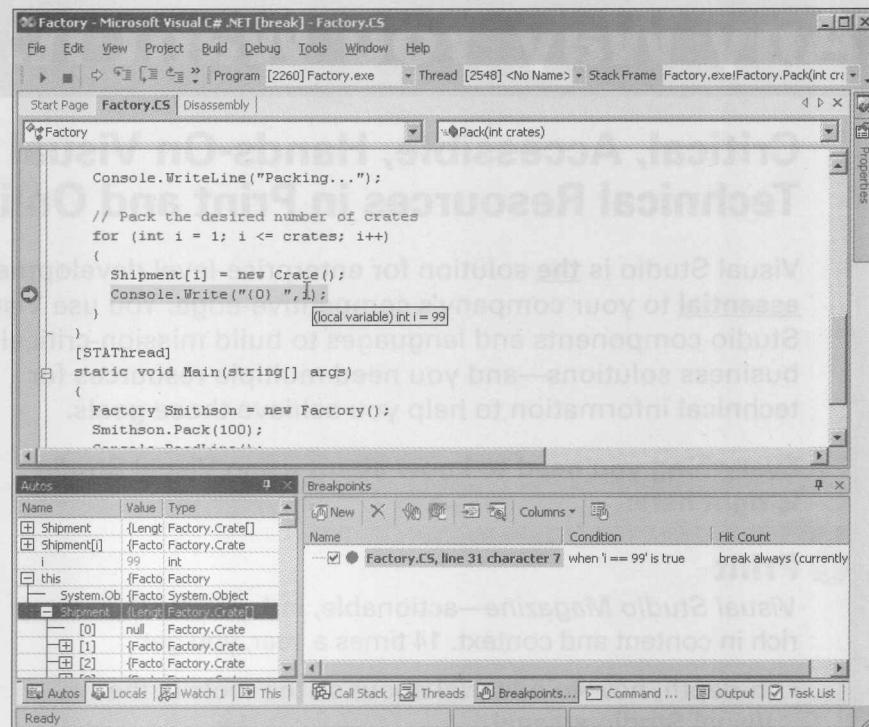


Figure 1 Use Debugging Windows to Diagnose Problems. When an error occurs in a program, elect to break it so you can see not only the line on which the error occurs, but the context of the error as well. What are the values of the variables? Are they what you expect? Does the code sometimes execute correctly? You can answer these questions quickly using the debugging windows.

When fixing a logic bug such as this one, check other related code for the same problem. Most of us are prone to making the same sorts of errors, and sometimes the most effective debugging technique is to understand your own vulnerabilities. The Factory example shows that each debugging tool is suited to different sorts of problems, and it takes some experience to figure out what each tool does and which sorts of problems each is best suited for.

Use the Watch windows to enter expressions you want to evaluate as your program is running in debug mode. As you step through or debug your code, the results update to show current values, and you can change the value of variables in the Watch windows. The four Watch windows allow you to group together related watch expressions.

Add a Watch Point

You can add expressions to the Watch window in three ways. You can right-click on any program variable and add it as a watch point or quick watch point. A watch point persists until you remove it. A quickwatch point is a one-time evaluation. However, you can create a persistent watch point from the QuickWatch window. Second, you can select and drag and drop from the code window into a Watch window. If the Watch window is closed and you drag over the tab, it expands and allows you to drop the variable as a watch in the window. Finally, you can drag and drop variables from the Command, Locals, and Autos windows into a Watch window. It would be nice if you could drag a variable into the Breakpoints dialog to create a breakpoint on the variable that's changing, but this feature is missing from the first release.

Everything You Need To Know, Any Way You Want It:

Critical, Accessible, Hands-On Visual Studio® Technical Resources in Print and Online

Visual Studio is **the** solution for enterprise-level development **essential** to your company's competitive-edge. You use Visual Studio components and languages to build mission-critical business solutions—and you need multiple resources for technical information to help you achieve those goals.

Everything you need to know about using Visual Studio is right here:

1 Print

Visual Studio Magazine—actionable, independent, and rich in content and context. 14 times a year, you get:

- In-depth, how-to editorial about all the tools in the Visual Studio arsenal
- Case studies, features, product reviews, code, and more

Get your **free trial issue now**—go to www.fawcette.com/getvsm/5C11,
Or call: 1-866-387-9275



2 Online

www.visualstudiomagazine.com builds on and expands the editorial from *Visual Studio Magazine*, and delivers unique content unavailable elsewhere, including:

- Free archives—thousands of pages of code, articles, tips, and more from back issues
- Discussion groups and expert advice

3 Weekly E-Newsletters

Free!

Extra content from the magazine and Web site—hot topics, code, opinions and more—delivered to your in-box every week, absolutely free. Sign up today at www.visualstudiomagazine.com

Visual Studio Information, Any Way You Want It.

FTP
FAWCETTE
TECHNICAL
PUBLICATIONS

These Visual Studio resources are brought to you by Fawcette Technical Publications, the leader in hands-on, how-to technical information for IT development professionals. www.fawcette.com • www.visualstudiomagazine.com • www.thedotnetmag.com

The Command window has two modes. Immediate mode is similar to the Watch window, and you can type in expressions. You can also type in any single line of code that's valid in the current context, allowing you to run methods, set and query the value of variables, and so on. Unfortunately, IntelliSense doesn't work in the Immediate mode.

In Command mode, you can evaluate some VS.NET commands with limited IntelliSense support; for example, type in ">Debug.ListPrograms" and you see this echoed:

```
>Debug.ListPrograms
Index Id      Process      Program
----- 
*1   1548  Factory.exe  Factory.exe
You can see what commands are available in the MSDN documentation under "Pre-defined Visual Studio Command Aliases" or by typing "alias" at the command prompt (see Additional Resources). You can also create your own aliases for commands you use commonly (see Additional Resources). IntelliSense lists commands even if they aren't available in the context, and it displays a message to that effect if you select an unavailable command. For example, choose this using IntelliSense:
```

```
>Debug.ApplyCodeChanges
```

You see this message: "Command 'Debug.ApplyCodeChanges' is not available."

Here's an example of the difference between Immediate and Command mode behavior. If you have a > prompt, you're in Command mode and can press Ctrl-Alt-I (or type "immed" and press Enter) to switch to Immediate mode, where the Command window displays "Immediate" in the caption. To go from Immediate mode to Command mode, type >cmd (or press Ctrl-Alt-A). The language you're using affects both the Command and Immediate windows. In the immediate mode in C#, type this and press Enter to see "8":

```
?3 + 5
```

In Command mode, you see the > prompt. Type this and press Enter to see "8" returned:

```
eval 3 + 5
```

Eval doesn't work in VB.NET; instead, you can use the ? for Debug.Print in both modes. Because ? is a debugger alias and not a language element, it works with both languages' Command and Immediate modes.

An example of another debugger alias is "Shell," which is an alias for Tool.Shell. Use this in either VB.NET or C# to run an external executable from the command window. For example, typing this starts an instance of Notepad:

```
>Shell notepad.exe
```

You can use this technique to run (and debug) another .NET EXE. To see an example of this, build an EXE from the Factory solution, then copy the Factory.exe to some directory; for simplicity, we've used the root folder of drive H:\. Now, start a new .NET session and open the Command window (Ctrl-Alt-A). Type this at the Command window:

```
>Shell H:\Factory.exe
```

Factory.exe has a bug in it, so the Just-In-Time Debugging window opens. It gives you the option to debug the Factory program. Once you do so, a dialog opens that lets you attach to the Factory process and debug the code. Try this same thing while at a breakpoint in some other .NET project. You'll be able to debug two applications at once.

Using breakpoints feels familiar, but you'll be glad to find additional features in the breakpoint dialog. You can make the breakpoint conditional on a variable value or on a hit count, and turn individual breakpoints on and off through a checkbox. Best of all, breakpoints now persist between debugging sessions, unlike in VB6.

Debug Subtle Problems

Of course, some bugs aren't solved as easily. The next example shows how debugging Windows.Forms events can present special challenges. Not only does the solution depend on the IDE debugging components, but it also highlights how important having experience using a language can be. All the code for this example is in VB.NET.

Here's the scenario: A data-entry form accepts a date and a name. Only the date is of interest. The date is validated to be after 1 January 2002, using a message box to indicate failure. The form works perfectly *most* of the time. However, users report that occasionally entered dates "just disappear" and they don't get prompted that the

C# • Identify the Error Using .NET's Debugging Windows

```
using System;
using System.Diagnostics;
using System.Text;

public class Factory
{
    public Crate[] Shipment;
    public class FruitLoop {}
    public class Box
    {
        FruitLoop[] FruitLoops;
        public Box()
        {
            FruitLoops = new FruitLoop[1000];
        }
    }
    public class Crate
    {
        Box[] Boxes;
        public Crate()
        {
            Boxes = new Box[144];
        }
    }
}

public void Pack(int crates)
{
    // Create a new shipment
    Shipment = new Crate[crates];
    Console.WriteLine("Packing...");

    // Pack the crates
    for (int i=1; i<=crates; i++)
    {
        Shipment[i] = new Crate();
        Console.Write("{0} ", i);
    }
}
[STAThread]
static void Main(string[] args)
{
    Factory Smithson =
        new Factory();
    Smithson.Pack(100);
    Console.ReadLine();
}
```

Listing 1 This code compiles without error, but an array bounds error occurs when it runs. Upon review, it seems to be packing the correct number of crates. You can identify this problem using the .NET debugging windows.

date is in the wrong year (see part of the VB.NET code in Listing 2; download the full version from the VSM Web site).

The first task is reproducing the bug. This is often difficult because developers don't do things exactly like users, or all the bugs would be fixed during development. In this case, the bug isn't reproduced until you know the specific keystrokes that cause it. Once the user tells you he or she has entered "20/8/2001" instead of "I'm putting in the 20th of August, 2001," you can conclude that the bug involves dates entered in European format.

A rule of thumb is to trust the youngest code the least, so check the debug statements first.

Once the bug is reproducible, it's time to start gathering more facts about the bug—frequently through hunches. You use the .NET Framework Trace and Debug classes to collect information. Although the members are similar, the classes have distinct purposes, so the compiler treats them differently from one another.

When a developer adds code to a class, the compiler adds a listener to the class that monitors the code for errors.

When a developer adds code to a class, the compiler adds a listener to the class that monitors the code for errors.

You use the Debug class during development. The compiler doesn't include the debug statements in Release builds (the type of build is selected in the Configuration Manager dialog found in VS.NET's Build menu). You use Trace for code you want to leave in the deployed application. Both Trace and Debug use "listeners," which are an output stream, such as the console, a file, and so on. The default listener for the Debug class is the debugger's output window. Trace code remains in the application, and you can add or remove listeners as needed.

Back to the example: When the bug is related to actions—entering a date, in this case—you place debug code in events. You should generally add a unique output line to all the event handlers that might be fired. In this case, a Debug.WriteLine statement outputs the current method name, but sometimes you'll also want to output a key parameter's value:

```
Private Sub FormatDate(ByVal sender As Object, _
    ByVal e As ConvertEventArgs)
    Debug.WriteLine("In Format Event")
    If Not e.Value Is DBNull.Value Then
        e.Value = String.Format("{0:d}", e.Value)
    End If
End Sub
```

VB.NET lets you add events by using either the Handles clause or the AddHandler method, so the best way to find all your event handlers is to search for both the Handles clause and the AddHandler method.

VB.NET • Users Report a Bug in This Simple Form

```
Option Strict On
Imports System
Imports System.Diagnostics
Imports System.Data
Imports System.Windows.Forms

Public Class Form1
    Inherits System.Windows.Forms.Form
    Private WithEvents m_dt As New dataTable()

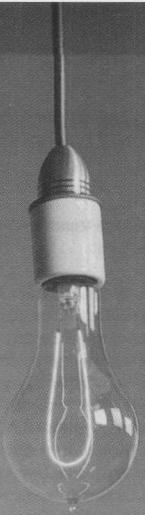
#Region " Windows Form Designer generated code "

Private Sub Form1_Load( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles MyBase.Load
    BuildData()
    Dim Binding As Windows.Forms.Binding = _
        New Windows.Forms.Binding("Text", _
        m_dt, "TestDate")
    AddHandler Binding.Format, _
        AddressOf FormatDate
    AddHandler Binding.Parse, _
        AddressOf ParseDate
    txtDate.DataBindings.Add(Binding)
    txtName.DataBindings.Add("Text", m_dt, _
        "TestName")
End Sub

Private Sub FormatDate(ByVal sender As _
```

```
Object, ByVal e As ConvertEventArgs)
If Not e.Value Is DBNull.Value Then
    e.Value = String.Format("{0:d}", _
        e.Value)
End If
End Sub
Private Sub ParseDate(ByVal sender As _
    Object, ByVal e As ConvertEventArgs)
    Debug.WriteLine( _
        Convert.ToDateTime(e.Value))
End Sub
Private Sub dataTable_ColumnChanging( _
    ByVal sender As Object, _
    ByVal e As DataColumnChangeEventArgs) _
    Handles m_dt.ColumnChanging
    If CDate(e.ProposedValue) < #1/1/2002# Then
        MessageBox.Show( _
            "Date must be after 1/1/2002")
        Throw New _
            System.ArgumentOutOfRangeException()
    End If
End Sub
Private Sub BuildData()
    m_dt.Columns.Add(New DataColumn( _
        "TestDate", GetType(String)))
    m_dt.Columns.Add(New DataColumn( _
        "TestName", GetType(String)))
    m_dt.Rows.Add(m_dt.NewRow)
End Sub
End Class
```

Listing 2 The form displays when the date entered is prior to 1/1/2002, and users complain that some dates "disappear." Try spotting the bug. Evaluating the scenario explained in this article, you're likely to resolve the bug faster with a conscious, or formal, approach to debugging.



how many people does it take to change the web?

macromedia®
DREAMWEAVER®
MX

See for yourself. With the all new Dreamweaver MX you can quickly and easily develop ASP.NET web applications. Be more productive with advanced design and code features including custom tags and code completion, DataGrid and DataList objects, and XML web services—all in a development environment familiar to web pros. Try Dreamweaver MX today and see what you can do now.

Download the free 30-day trial at
www.macromedia.com/go/dwmxd



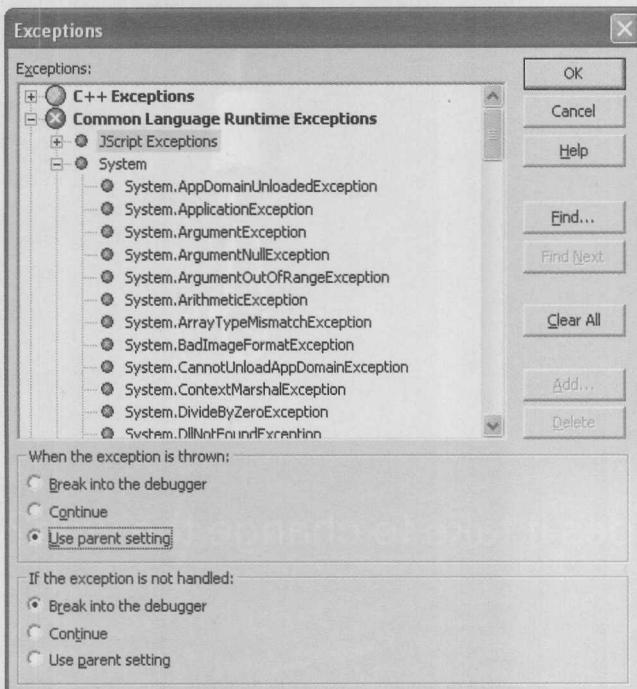


Figure 2 Focus Your Exception Behavior. The Exceptions dialog can catch exceptions that are difficult to catch any other way. In fact, this technique is so powerful that it breaks on exceptions in the .NET Framework. Expanding the tree and using the hierarchy designed into your own exceptions and Framework exceptions allows you to focus your exception behavior as narrowly or as broadly as you need.

The output from the next debugging session shows that you're calling only the parse event:

In Parse Event

This is surprising, because both the Parse and the ColumnChanging handlers have debug statements that should have ex-

ecuted. A rule of thumb is to trust the youngest code the least, so check the debug statements first. If these statements look okay, maybe you didn't attach the ColumnChanging handler correctly. A suspicion is something that can be tested, and you can phrase it as a hypothesis. A hypothesis is a hunch phrased in a way that allows you to test it. Focus on disproving instead of proving your hunch because test results can be consistent with multiple explanations. A failed test tells you that you're on the wrong track, while a successful test doesn't necessarily tell you that you're on the right track.

Test Your Hypothesis

Add a breakpoint on the AddHandler line and check that it's being run. When the program hits this breakpoint, your hypothesis is wrong and you need to step back to the fact-finding stage and find a new hunch to explore. It's natural to think that disproving your hunch is wasted time. But you don't waste time providing and testing an incorrect solution, and you know precisely why you abandoned this direction—if you take good notes.

Research known behavior in addition to exploring your code. When debugging events, it helps to understand that when an exception occurs in a Windows.Forms event, remaining events are generally not fired. This behavior is logical because it mimics the behavior of multicast delegates, and some Windows.Forms sequential events use multicast delegates. Exceptions that are raised in events aren't necessarily raised in your code.

Based on this information, a reasonable hypothesis is that an exception is occurring in or after the Parse event, which is suppressing later events, including ColumnChanging. From a practical perspective, testing a hypothesis depends on the tools you have available. The hypothesis that events are being suppressed is testable because .NET has the Exceptions dialog tool that lets you request breaking into the debugger on all exceptions. If the application breaks into the debugger, there's support for the hypothesis (though not proof).

Place a breakpoint at the start of the Parse event. When you hit this breakpoint, select Debug | Exceptions from the menu to display the Exceptions dialog (see Figure 2). Select "Common Language

Use the Scientific Method

The scientific method is useful for everyday problem solving. Use the method to *observe* a problem, *investigate* when it happens and how it looks, *hypothesize* causes, *experiment* to test hypotheses, and *analyze* alternatives. Apply the method iteratively to complex problems to refine your understanding until a solution is apparent.

1. *Observe* a problem to develop a problem statement. "Dates don't work" is vague, whereas "I can enter dates for last year, and that's wrong" is clear.
2. *Investigate* the problem. Observe when it happens and how it looks, and weed out irrelevant observations by listing the steps necessary to reproduce the problem.
3. *Hypothesize* one or more explanations. Pick the most likely, or the easiest, to test first.
4. Test your hypothesis with an *experiment* designed to *disprove* it. If you can't disprove your hypothesis, you're not done. You might

have to rule out other possible explanations.

5. *Analyze* the results. Was your hypothesis wrong, or is the solution clear? Consider alternatives: Patch, rewrite, workaround, or do nothing if the cure is worse than the disease. Use the project budget, schedule, and scope to determine the best course of action.

The methodology is compartmentalized in theory; frequently, in practice, it's not. Even simple debugging includes all these steps, although they can be transparent. Notice the steps don't include implementing a solution. For a sufficiently complex solution, approach the fix the same as you would writing any maintenance code. If you use source code control, you can roll back the changes you made while experimenting, make the correction, then test only the fix. While this seems like overkill, it can avoid the difficulty of tracking down a bug that's introduced as a side effect of making other changes.

Add Dynamic Maps to Your Applications



By building mapping and geographic information system (GIS) capabilities into your applications, you can help users to spot patterns and trends in their data that might otherwise go unnoticed. The result is better, more informed decision making.

With ESRI® MapObjects® software, you can quickly and easily add mapping capabilities to your applications. MapObjects includes nearly 50 embeddable mapping components that you can use in standard Windows development environments such as Visual Basic®, Visual Basic for Applications (VBA), Visual C++®, Delphi™, Borland® C++ Builder, Visual FoxPro®, and PowerBuilder®.

MapObjects supports advanced geocoding, geometric functions, on-the-fly map projection, GPS management, street map data, CAD drawing files, a wide variety of image formats, and much more.

Download a free evaluation version of MapObjects from www.esri.com/mapobjects.

MapObjects®
Embeddable Mapping and GIS Components



1-888-332-2320

www.esri.com/mapobjects • info@esri.com



Visual Studio Developer Conference

Orlando September 14-19



Get Top Performance with Visual Basic .NET

Access Data with ADO .NET

Scale, Secure and Extend Your XML Web Services

Build Rich Web Apps with .NET Enterprise Servers



Get Maximum Reusability with Templates and Inheritance

Incorporate Client-Side Code with ASP.NET Server Controls

Create Better Design Experiences

Use Best Practices for Data Access



Attend All 6 Days & Enjoy Gold Passport Benefits.

Get all the benefits of being a Gold Passport Holder. One convenient package offers open access to everything at VS Live!, including 4 complete conferences—VBITS, C# Live!, ASP Live! and SQL Live!—and 6 complimentary full-day pre- and post-conference workshops.



Time and Budget a Concern?

Purchase the ticket tailored to your specific development focus. Whether it's ASP Live!, C# Live!, or SQL Live!, you can attend for as little as \$595 when you sign up by August 7th!



Everyone Attends the .NET Focus Day for Free!

When you purchase any VS Live! package—either the Complete Gold Passport or the VBITS/C# Live!/ASP Live!/SQL Live! ticket—you are automatically invited to attend a full day of all-new sessions presented by leaders within the Microsoft Development Team.

Put .NET to Work in 4 Unique Events



**Step Through Component
Architecture and Distributed App
Development**

**Dive Into Advanced C# Language
Features, Metadata and Reflection**

**Balance Client-Side vs. Server-Side
Operations in Web Applications**

**Tackle Multithreading and
Multiprocessing Issues in .NET**



Secure Your SQL Server Data

**Design For, Improve, and Monitor
Performance**

**Connect and Manage The Connection
Pool from ADO .NET**

Get High Performance Data Loading

**Register by August 7th Save \$200!
Call Today for Details! 800-848-5523**

www.vslive.com/orlando

Microsoft, MS, MSDN, SQL Server, Visual Basic, Visual C++, C#, Visual Studio and Windows are registered trademarks of Microsoft Corporation. VSLive! and *Visual Studio Magazine* are trademarks of Fawcette Technical Publications, Inc. Visual Studio is used by Fawcette Technical Publications, Inc. under license from Microsoft. All other trademarks are property of their respective owners.



Runtime Exceptions" from the tree and "Break into the debugger" for "When the exception is thrown." If you're interested only in specific exceptions or exception families, expand the tree and refine which exceptions break into the debugger. When run, the application breaks into the debugger, allowing you to capture an exception you wouldn't see outside the Parse event.

Even if the hypothesis appears correct, it doesn't solve the problem because it might open new questions. For example, if the exception is causing other events not to fire, why is the exception being raised? This question leads you to hone in on this line of code:

```
Debug.WriteLine(Convert.ToDateTime _  
    (e.Value).ToString())
```

Now, you might wonder whether Convert.ToDateTime causes an exception on European dates. This illustrates how important test conditions are in debugging, because the answer depends on the users' regional settings. At this point, your testable hunch and your hypothesis is that Convert.ToDateTime causes an exception on European dates if the regional settings are U.S.

You could remove the line because it was a mistake to leave it in after your earlier debugging. However, that leaves an underlying problem. Notice that the application behavior is inconsistent. The bug occurs if the intended day is greater than 12, and the application changes the date if it's valid as a U.S.-format date.

Additional Resources

- *Code Complete: A Practical Handbook of Software Construction* by Steve McConnell [Microsoft Press, 1993, ISBN: 1556154844]
- "Pre-defined Visual Studio Command Aliases": <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsintro7/html/vxgrfpredefinedcommandlinealiases.asp>
- "Creating Custom Aliases for Visual Studio Commands": <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsintro7/html/vxtskcreatingcustomcommandlinealiases.asp>

In this case, the solution is partly programming (remove the unintentional debug statement) and partly a matter of stepping back into design. Once you understand the problem clearly, you might need to sit down with the customer to figure out the best solution. Consider all alternatives (see the sidebar, "Use the Scientific Method").

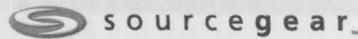
You've seen several important debugging concepts and explored some interesting debugging tools in the VS.NET IDE. In particular, this article's examples show that a single bug might point to more than one underlying problem. Similarly, you might need to investigate alternative solutions because they can vary significantly in cost and effort. Benchmark alternatives to help identify the best one. Be cautious about making changes that affect the public interface of an application or component without considering how the change impacts the original design.

Remember that building software is an iterative process. Developers and managers rely frequently on their own judgment when choosing a solution, and often that's appropriate. However, if a solution to a bug affects the user, it's a design question and will at a minimum require updating the project's documentation. A particularly difficult or nonobvious bug might require adopting a more formal debugging approach. Adopt the scientific method consciously as a way to prompt yourself to find any unsuspected assumptions or overlooked observations. **VSM**

Remote access to your SourceSafe development team.

SourceOffSite® Collaborative Edition combines source code control with bug tracking, chat, discussion groups and operation notifications to create a rich collaborative environment for users of Microsoft® Visual SourceSafe® (VSS) 6.0.

Download free, fully-functional, 30-day trial versions at www.sourcegear.com



www.sourcegear.com 1-877-356-0106 info@sourcegear.com

SourceOffSite is a registered trademark of SourceGear Corporation. Microsoft and Visual SourceSafe are registered trademarks of Microsoft Corporation. Must be a user of Microsoft Visual SourceSafe.



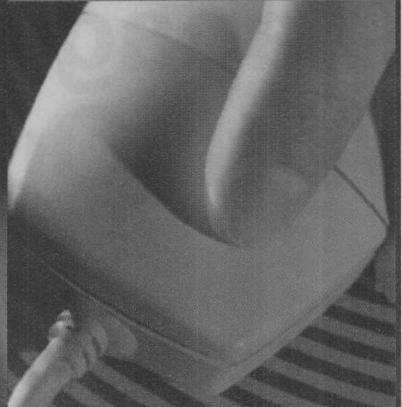
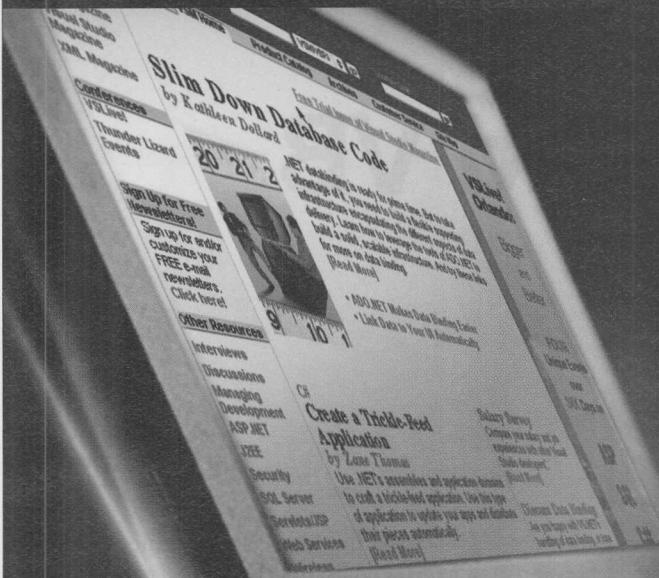
Nancy Folsom is an independent consultant based in Seattle. She's a Microsoft MVP, is the author of *Debugging Visual FoxPro Applications* (Hentzenwerke Publishing), and is active in the Seattle FoxPro User Group. Reach her at nancy@microsoft.org.



Kathleen Dollard is an independent developer based in Fort Collins, Colo. She's a Microsoft MVP, enjoys speaking at conferences such as VSLive!, and is active in the Denver Microsoft Visual Studio .NET User Group. Reach her at kathleen@microsoft.org.

Visual Studio® Technical Info, Online

FTPOnline



Click



Go To www.visualstudiomagazine.com and get how-to, hands-on Visual Studio technical information to help you program better and faster.

- Discussion groups
- Exclusive Web casts
- Codes, tips, advice and more
- In-depth coverage of key Microsoft technologies, including Visual Studio .NET, C#, C++, SQL Server, and more!

Connect



Plus, FREE weekly e-newsletter. Sign up now!



visualstudiomagazine.com

© Fawcette Technical Publications, Inc. Visual Studio is a registered trademark of Microsoft. Visual Studio Magazine is a trademark of Fawcette Technical Publications, Inc. All product names herein are the properties of their respective owners.

Code

Manage Contacts With Outlook and VB

Program Microsoft Outlook's object model from your VB6 app to manage contact lists.

by Stan Schultes

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**
- Other:**

Outlook 2000 or 2002

ADO

You can extend Microsoft Office's functionality by reusing portions of Office apps through exposed ActiveX objects. You can write Visual Basic for Applications (VBA) code inside individual Office apps, creating "macros" you then add to buttons and menus. And you can create add-ins for extending each Office app in a consistent manner. Best of all—at least for VB developers—you can access the Office suite's object model from external apps.

I'll demonstrate this extensibility by showing you how to collect e-mail addresses on your Web site and build a contacts list your marketing department can use from Outlook. My ContactsUtility sample app shows you how to program Outlook by reading contact info from an Access database and managing entries in a contacts folder (download the app from the VSM Web site; see the Go Online box for details).

ContactsUtility lets you add/update contacts in a folder, and list or delete existing contacts (see Figure 1).

You gain access to Outlook's objects by adding a reference in your VB app to Outlook's Object Library. You can then program against the exposed object model. Go to Project | References, and set a reference to the Microsoft Outlook 9.0 Object Library for Outlook 2000 (use the Outlook 10.0 reference for Office XP). The ContactsUtility has the reference set already; open the sample app in VB6 and follow along.

For the sample contacts, use the Suppliers table from the Northwind sample Access database. The Suppliers table doesn't have an Email field by default, so I added one and created fake e-mail addresses for the supplier contacts by combining the contact and company names (see the readme file for details). This database is included in the download as NWindEmail.mdb.

ContactsUtility is organized as a single form named frmMain. The form includes three class modules: CDatabase for database access, COutlook for the Outlook interface, and CSettings to read and save configuration settings in an application INI file. It's good practice to use classes to separate your app's functionality into distinct units, because it aids in developing and understanding the app, and it eases later maintenance chores.

The simplest of the three classes is CSettings. Two basic methods, GetSettingString and SaveSettingString, read and write settings from the app's INI file, which you find in the app's EXE path. Two other methods, Get.AppSettings and Save.AppSettings, load and save the size and screen location of frmMain, so you can place the

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209GS Download the code for this article, which includes the ContactsUtility sample application that uses Outlook's object model to maintain Outlook contacts from an Access database table.

Discuss

VS0209GS_D Discuss this article in the "Classic" VB forum.

Read More

VS0209GS_T Read this article online. It includes two extra code listings that show you how to add and update Outlook contacts.

VB0007GS_T Getting Started, "Use SQL With ADO," by Stan Schultes

VB0009GS_T Getting Started, "Make Application Settings Easy," by Stan Schultes

VB9911SS_T "Create COM Add-Ins With VBA in Office 2000" by Stan Schultes

form on startup where it was the last time it ran. You might want to include this user-friendly feature in your application toolkit. On app startup, load settings and initialize the CDatabase class in frmMain:

```
Private WithEvents moDB As CDatabase
Private Sub Form_Load()
    'handles form startup
    LoadSettings
    Set moDB = New CDatabase
End Sub
```

The LoadSettings sub calls the CSettings.GetAppSettings method to read settings from the [Settings] section of ContactsUtility.ini:

```
[Settings]
WindowState=0
FormLeft=1650
FormTop=1650
FormHeight=5145
FormWidth=6450

[Database]
Connect=Provider=Microsoft.Jet.OLEDB.4.0;
DBName= Data Source=NWindEmail.mdb
EmailTable=Suppliers

[Outlook]
ContactsFolder=Personal
Folders.NwindContacts
```

You also create an instance of the CDatabase class and store it in the form-level variable moDB. In the QueryUnload event, you do the reverse by saving settings and setting the moDB variable to Nothing to dispose of the object.

Keep the Progress Bar Up to Date

Use the WithEvents keyword in the declaration moDB as CDatabase. This tells VB you want to receive events in the form code generated by the CDatabase class at run time. Three events control a progress bar (prgContacts) on the form—one event at the start of an operation, another at the end, and one more as you process each contact item. Handle the per-contact event and update the progress bar within frmMain like this:

```
Private moContactItems As Long
Private Sub moDB_UpdateOperation _
    (ByRef Item As String, _
     ByRef CurrentItem As Long)
    If Len(Item) Then lstContacts.AddItem Item
    prgContacts.Value = CurrentItem / _
        moContactItems * 100
    DoEvents
End Sub
```

Add the returned Item string to frmMain's lstContacts listbox if it has a value, and update the progress bar value as a percentage of its

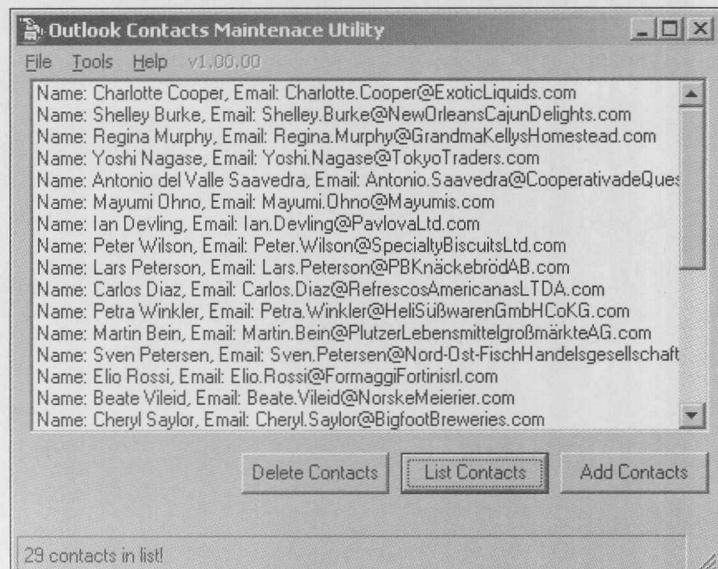


Figure 1 Access Outlook Contacts Programmatically. The ContactsUtility main form demonstrates how to access Outlook contacts from within your VB6 application. ContactsUtility shows a progress bar, driven by application-generated events as it adds, lists, or deletes contacts. The contacts information comes from an Access database, with the whole application driven from an INI file defining configuration parameters.

total, based on the starting value you store in moContactItems when the contact operation begins.

The other two utility classes have a relationship in ContactsUtility. The CDatabase class contains all ADO code that reads and writes the sample database. You call three CDatabase methods from frmMain. These correspond to the three ContactsUtility functions: AddContacts, DeleteContacts, and ListContacts. These methods in turn call COutlook methods to access the contact data inside Outlook. You don't call the COutlook methods directly from the main form. In this sense, COutlook is *contained* by the CDatabase class.

CDatabase handles the interface to the NWindEmail database through ADO. Set a reference in ContactsUtility to the Microsoft ActiveX Data Objects 2.5 (or later) Library. You can find the ADO Connect string for the database in the app's INI file in the [Database] section. Keep variable settings such as database connect strings in an INI file to give your app flexibility over time. In this case, you can change the database location without having to rebuild the app.

COutlook methods raise events as they progress through their tasks: StartOperation, UpdateOperation, and EndOperation. For example, the ListContacts method raises StartOperation at the beginning of the listing process, returning the total number of contacts found. As your program iterates through each contact, ListContacts raises UpdateOperation, which returns the contact name and current contact count. When it finishes, ListContacts raises the EndOperation event, which returns a status string. The CDatabase class method passes each of the raised events right through to the calling routine.

The real Outlook programming work in ContactsUtility occurs within the three public COutlook methods. When you create an instance of the COutlook class, its Class_Initialize event creates an

| | | |
|----------------|-------------|--------------|
| ON TIME | 6155 | 11:05 |
| ON TIME | 927 | 11:10 |
| ON TIME | 1411 | 11:20 |
| DELAYED | 2643 | 11:25 |
| ON TIME | 466 | 11:35 |
| ON TIME | 842 | 11:40 |
| DELAYED | 1209 | 11:45 |
| ON TIME | 1744 | 11:55 |



It has been clear for some time now that the future will be a world in which everything is connected to everything, seamlessly, automatically, and securely, via XML Web services. What has not been clear is how that future will be built. Introducing Microsoft® Visual Studio® .NET, the first toolset built from the ground up for XML Web services development. Just one keyword—WebMethod—

turns your existing application into a Web service, and with new Web Forms you can create Internet applications using the same “drag-and-drop, double-click to write code” technique made famous


© 2002 Microsoft Corporation. All rights reserved. Microsoft, Windows, Visual Basic, Visual C++, Visual C#, Visual Studio, the Visual Studio logo, and the .NET logo are either registered trademarks or trademarks of Microsoft

81A

74

Introducing
Visual Studio .NET

62

31

77A

49

66

The world takes
a little step
into the future.

LONDON

COPENHAGEN

MARS

MILAN

TOKYO

MONTREAL

NEW YORK

BUENOS AIRES



by Microsoft Visual Basic®. Plus, the whole process of building Windows®, Web, and mobile applications gets faster and easier with the new Integrated Development Environment (IDE), which gives you one toolbox, one debugger, one task list, one XML designer, and more. And within the IDE you get support for over 20 languages, including Visual Basic .NET, Visual C++®.NET, and new Visual C#™.NET, so you can leverage your existing skills and code across development teams and projects.

Start building the future: visit msdn.microsoft.com/vstudio Software for the Agile Business.

Microsoft®

instance of the Outlook Application object, then calls a private Init routine (see Listing 1):

```
Private Sub Class_Initialize()
    Set m_oOL = New Outlook.Application
    If Not Init Then Set m_oOL = Nothing
End Sub
```

Use the Outlook Application Object

The Outlook Application object functions as the top-level "controller" you use to program objects exposed by Outlook. The Init routine creates an instance of the Outlook Namespace object:

```
Set m_oNS = m_oOL.GetNamespace("MAPI")
```

Init then reads the app's INI file, looking for a key named ContactsFolder in the [Outlook] section. If ContactsFolder exists, parse the period-delimited list and initialize the Folder object. If it doesn't exist, set the Folder object to the user's default Contacts folder and set a class-level flag:

```
Set m_oFold = m_oNS.GetDefaultFolder _
    (olFolderContacts)
mbDefaultFolder = True
```

VB6 • Set Up Outlook for Programmatic Access

```
Private Function Init() As Boolean
Dim oSet As New CSettings
Dim sFold As String
Dim sFolders() As String
'error handling omitted for space ...
Set m_oNS = m_oOL.GetNamespace("MAPI")
sFold = oSet.GetSettingString("Outlook", _
    "ContactsFolder", "")
If Len(sFold) Then
    mbDefaultFolder = False
    sFolders = Split(sFold, ".")
    If UBound(sFolders) = 2 Then
        Set m_oFold = m_oNS.Folders _
            (sFolders(0)).Folders(sFolders(1)) _
            .Folders(sFolders(2))
    ElseIf UBound(sFolders) = 1 Then
        Set m_oFold = m_oNS.Folders _
            (sFolders(0)).Folders(sFolders(1))
    Else
        Set m_oFold = m_oNS.Folders _
            (sFolders(0))
    End If
Else
    'use default contacts folder
    Set m_oFold = m_oNS.GetDefaultFolder _
        (olFolderContacts)
    mbDefaultFolder = True
End If
Init = True
End Function
```

Listing 1 The private Init function sets up the Outlook objects required for programmatic access to contact lists. The Outlook folder you use for contacts comes from the application's INI file as a period-delimited string. Parse the folder string and set a flag if you use your default contacts folder, because some code differs in other COutlook class methods.

Once you set the Folder object, you can access contacts within that folder. The sample app provides three choices for contacts folders, starting with your default contacts folder. Then there's a public folder you can access on your Exchange server. Both of these choices require you to be connected to your Exchange server. You can also access contacts in a personal folder, which lets you test your code offline. You create personal folders with Outlook's File | New | Personal Folders File menu item. See the Outlook documentation for working with personal folder files.

The AddContacts method merits a closer look (download Listing 2). The CDatabase class passes an ADO recordset as a parameter to this method. This introduces *coupling* between the classes, where one class has knowledge of how another works. Normally, you want to avoid this, and you do so by creating a COutlook property for each database field, filling the properties in CDatabase and creating contacts within COutlook using those properties. However, in my simple sample app, the recordset provides a convenient shortcut that reduces the amount of programming required.

AddContacts works by looping through the recordset of contact entries and searching the Outlook contacts list for a match by e-mail address. If AddContacts doesn't find a match, you add the contact. If the contact exists already, you update it.

Although this example matches by e-mail address, you need to think about an appropriate match for your application. If the database entries comprise an e-mail list, every contact most likely has a unique e-mail address, so searching by e-mail makes sense. If the e-mail address in your app is optional, however, you might not have an entry to match by. Also, remember that unlike e-mail addresses, names are not unique. Many people in the world share exactly the same name.

In AddContacts, start by signaling the total number of records to process with the StartOperation event:

```
lListCount = RS.RecordCount
RaiseEvent StartOperation(lListCount)
```

Set a filter on the items in the contacts list to avoid any Distribution List items users might create manually in the Contacts folder:

```
Dim oItems As Outlook.Items
Set oItems = m_oFold.Items.Restrict _
    ("[MessageClass] = 'IPM.Contact'")
```

Update Recordset Contacts

Search the Contacts folder, and add or update each contact in the recordset as appropriate:

```
Dim oContact As ContactItem
sFilter = "[E-mail] = "" " & RS("Email") & """
Set oContact = oItems.Find(sFilter)
If oContact Is Nothing Then
    sUpdate = AddContact(RS)
Else
    sUpdate = UpdateContact _
        (RS, oContact)
End If
```

New XML Web Site and Free E-Newsletter!

Maximize your XML Insights!

1

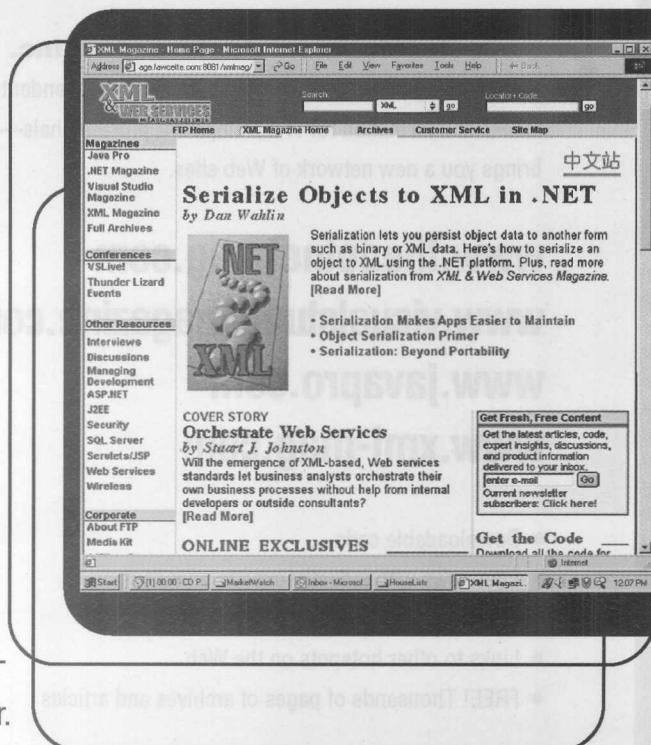
Go Online:

Go to www.xml-mag.com for the in-depth, how-to XML information you need: **code, articles, archives, case studies, interviews, and more.**

2

Free Weekly E-mail News!

Subscribe to **XML Insight newsletter**—the new, weekly, FREE e-mail newsletter. Get practical, hands-on articles, tips and enterprise-level solutions delivered to your in-box. It's easy, it's fast, it's free. Subscribe at www.xml-mag.com.



Brought to you by the publishers of

**XML
&
WEB SERVICES**
MAGAZINE

www.xml-mag.com

FTPOnline



FTPOnline—technical sites for IT development professionals

Get the technical information you need—when you need it.

Fawcette Technical Publications, Inc.

—the leading provider of hands-on, how-to, independent technical information for IT development professionals—brings you a new network of Web sites.

www.thedotnetmag.com
www.visualstudiomagazine.com
www.javapro.com
www.xml-mag.com

- Downloadable code
- Interviews with industry experts and visionaries
- Overviews of the newest technologies
- Links to other hotspots on the Web
- FREE! Thousands of pages of archives and articles

Free E-Mail Newsletters!

Pick your topic, decide how much technical information you want, and we'll send it to you every week, FREE.

Sign up today at www.fawcette.com

- .NET Insight
- Java Insight
- XML & Web Services Insight
- Web Design & Development Insight

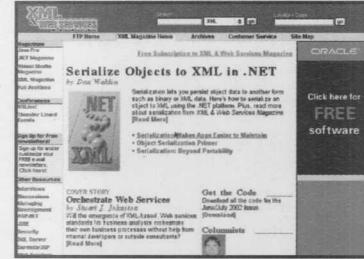
Go There Today:

www.fawcette.com

FTPOnline

© 2002, Fawcette Technical Publications, Inc.

All product names herein are the properties of their respective owners.



Start by setting a reference to a new contact in the AddContact routine (download Listing 3):

```
If mbDefaultFolder Then
    Set oContact = m_oOL.CreateItem _
        (olContactItem)
Else
    Set oContact = m_oFold.Items.Add
End If
```

You use different mechanisms to add a contact depending on whether you're adding to the default folder (using the CreateItem method) or to another folder (where you use the Add method of the Folder's Items collection). You then set the contact's properties and call the Save method to save the new contact:

```
With oContact
    If Len("") & RS("ContactName") =
        Then .FullName = Trim("") =
            & RS("ContactName"))
    'Other contact fields omitted
    .Save
End With
```

The empty string concatenated with a recordset field handles Null values in the database. A concatenation between an empty string and a Null value produces an empty string. This doesn't cause errors in VB functions (such as Len) that would throw an error if you passed them a Null value. The UpdateContact routine resembles AddContact, except it checks each field for a change. If it finds one, it updates the contact record using the same Save method.

When you program Outlook 2000 or Outlook 2002 object models, security dialogs pop up when you access your contacts lists or try to send e-mail. Microsoft introduced these "features" into Outlook between Office 2000 SP1 and SP2 to protect you from malicious programs accessing your address books and sending e-mail without your knowledge. Unfortunately, you don't have a lot of options for dealing with these intrusive behaviors (see Additional Resources for Microsoft's explanations).

The ContactsUtility sample application should get you started building apps that manipulate contacts in Outlook. Next, you might want to explore how to create and maintain distribution lists to help manage your e-mail contacts. ContactsUtility exposes only the tip of the Outlook object model iceberg; I encourage you to explore

further and see what you can find to enhance your own apps with built-in Office functionality. **VSM**

Stan Schultes is a Sarasota, Fla.-based Web and enterprise application architect and developer, and is an MCP in VB. Stan is a contributing editor for *VSM* and writes regularly for the magazine. Visit his Web site at www.vbnetexpert.com for online code

demos, updates, and other information. Reach Stan at stan@vbnetexpert.com.

Additional Resources

"Developer Information About the Outlook E-mail Security Update" (Q262701): <http://support.microsoft.com/directory/article.asp?id=kb;en-us;q262701&>



Show Your Sharper Edge

| NorthWind Products | | | | | | | |
|---|---------------------|-----------------|-------|----------|----------|-------------------------------------|--------------|
| CategoryName | ProductName | QuantityPerUnit | Price | In Stock | On Order | ReorderLevel | Discontin... |
| Category : Beverages | | | | | | | |
| Category : Condiments | | | | | | | |
| Condiments Aniseed Syrup | 12 - 550 ml bottles | \$10.00 | 13 | 70 | 25 | <input type="checkbox"/> | |
| Condiments Chef Anton's Cajun Seasoning | 48 - 6 oz jars | \$22.00 | 53 | 0 | 0 | <input type="checkbox"/> | |
| Condiments Chef Anton's Gumbo Mix | 36 boxes | \$21.35 | 0 | 0 | 0 | <input checked="" type="checkbox"/> | |
| Condiments Genen Shouyu | 24 - 250 ml bottles | \$15.50 | 39 | 0 | 5 | <input type="checkbox"/> | |
| Condiments Grandma's Boysenberry Spread | 12 - 8 oz jars | \$25.00 | 120 | 0 | 25 | <input type="checkbox"/> | |
| Condiments Gula Melaka | 20 - 2 kg bags | \$19.45 | 27 | 0 | 15 | <input type="checkbox"/> | |
| Condiments Louisiana Fiery Hot Pepper Sauce | 32 - 8 oz bottles | \$21.05 | 76 | 0 | 0 | <input type="checkbox"/> | |
| Condiments Louisiana Hot Spiced Okra | 24 - 8 oz jars | \$17.00 | 4 | 100 | 20 | <input type="checkbox"/> | |
| Condiments Northwest Cranberry Sauce | 12 - 12 oz jars | \$40.00 | 6 | 0 | 0 | <input type="checkbox"/> | |
| Condiments Original Frankfurter grüne Soße | 12 boxes | \$13.00 | 32 | 0 | 15 | <input type="checkbox"/> | |
| Condiments Sirop d'étable | 24 - 500 ml bottles | \$28.50 | 113 | 0 | 25 | <input type="checkbox"/> | |
| Condiments Vegie-spread | 15 - 625 g jars | \$43.90 | 24 | 0 | 5 | <input type="checkbox"/> | |
| Units on order: Aniseed Syrup, Louisiana Hot Spiced Okra | | | | | | | |
| Category : Confections | | | | | | | |
| Confections Chocolade | 10 pkgs. | \$12.75 | 15 | 70 | 25 | <input type="checkbox"/> | |
| Confections Gummibärchen | 100 - 250 g bags | \$31.23 | 15 | 0 | 0 | <input type="checkbox"/> | |
| Confections Maxilaku | 24 - 50 g pkgs. | \$20.00 | 10 | 60 | 15 | <input type="checkbox"/> | |
| Confections NuNuCa Nuß-Nougat-Creme | 20 - 450 g glasses | \$14.00 | 76 | 0 | 30 | <input type="checkbox"/> | |
| Confections Pavlova | 32 - 500 g boxes | \$17.45 | 29 | 0 | 10 | <input type="checkbox"/> | |
| Confections Schoggi Schokolade | 100 - 100 g pieces | \$43.90 | 49 | 0 | 30 | <input type="checkbox"/> | |
| Confections Scottish Longbreads | 10 boxes x 8 pieces | \$12.50 | 6 | 10 | 15 | <input type="checkbox"/> | |
| Confections Sir Dordogne Marmalade | 20 soft boxes | \$21.00 | 40 | 0 | 0 | <input type="checkbox"/> | |
| Sweet and savory sauces, relishes, spreads, and seasonings | | | | | | | |
|  <p>Name: Chef Anton's Gumbo Mix Per Unit: 36 boxes Price: \$21.35 In Stock: 0 On Order: 0</p> | | | | | | | |

- **ADO, DAO Data Binding**
- **Unbound mode: Event driven or using interfaces**
- **Variety of cell edits types: single and multi-line edit box, action button, check box and combo box**
- **Supports Alpha blending and gradient fills**
- **Implements a stage driven, custom draw mode you can intercept and replace one or more stages in control's paint cycle**
- **Data highlighting using styles**
- **Odd-Even rows highlighting**
- **Preview panel allows you to show contents of one column inside the preview pane**
- **Single or multiple column sorting**
- **Outlook style grouping and Group Calculations: Users can select one or more columns and group rows based on values in selected columns**
- **Frozen Rows and Columns**
- **Supports Print and Print Preview using Data Dynamics' Active Reports Viewer Control (included)**
- **Natively supports export to Excel worksheets. Excel not required.**

614-895-3142 DATA DYNAMICS Download Free Evaluation Copy from our website!

Fax 899-2943 www.datadynamics.com

VISUAL STUDIO MAGAZINE • SEPTEMBER 2002 • www.visualstudiomagazine.com

41

Deploy WinForms on the Web

Deploy and update .NET WinForms on the Web as easily as HTML-based Web apps—and still maintain security.

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209DT Download the code for this article, which includes the business DLL, two apps that call it, an app that uses Windows controls, and a DLL with WinForms inside.

Discuss

VS0209DT_D Discuss this article in the VB.NET forum.

Read More

VS0209DT_T Read this article online. It includes two extra sidebars, "Achieve Security Nirvana With Key Files and Smart Cards" and "Enable the Execution of Signed .NET Apps."

VS0207ZT_T "Create a Trickle-Feed Application" by Zane Thomas

You can marry the rich user experience of VB applications with the easy deployment and maintenance of Web apps by "Web-deploying" .NET WinForms apps. .NET lets you do this by bringing Web deployment to nearly conventional Windows apps.

An article in a previous issue showed you how to create trickle-feed apps by using file shares for downloading (see the Read More section in the Go Online box for details). My approach differs by using HTTP and Internet Explorer for downloading, and by relying on built-in .NET mechanisms for versioning, caching, and instantiation of classes. You can leverage your current VB knowledge to develop flashy, high-performance, productive apps. Your apps can use Web services to get data from the server—a more efficient technology than bringing back whole pages.

For example, a home banking app would deliver only your balance rather than a big page with your balance lost somewhere in the middle. Such an app could even have a private directory to save local data if necessary. Use bandwidth efficiently and produce a snappy app by downloading all the app's pieces (EXEs and DLLs) under demand and caching them pretty much the same way as Web pages. You can even work offline without the Internet connection if your app doesn't need server data. You can start a thread to download the app's components in the background while the user works on something downloaded previously. Try doing that with your regular Web app.

All this bliss does have a price. You need the 20 MB .NET Framework installed on each client, making this basically an intranet/extranet solution, at least for now. You can deploy pretty much any .NET WinForms app through a link on a Web page, regardless of the app's size.

However, I recommend dividing it into one EXE and several DLLs so each part can be downloaded on demand, much like an HTML-based app that's downloaded page by page. For example, a 500K app could become a 50K EXE and nine 50K DLLs. The latter might consist of "business" DLLs without user interface code, or DLLs that instantiate WinForms or Windows control DLLs with UIs of their own.

Be sure to give all assemblies "strong names" by signing them digitally. A strong name includes not only the module name (usually the EXE/DLL filename without the extension), but also a version number, the digital signature, and a culture (which can be neutral). If any of these things change, you have a new strong name and the .NET runtime regards the resulting file as different. It is possible to download a simple Windows app without using digital signatures, but you'll have trouble when you use multiple files or try to adjust security.

Create a Sample App

So much for theory—now for practice. I'll show you how to create digital signatures, a DLL, and an EXE, then deploy them over the Web. Type this code at the command prompt to create a "key pair" file with both a private and a public key:

```
sn -k MyKey.snk
```

This command creates a 596-byte file. Treat it with care. Anyone who obtains this file will be able to sign files in your stead. Put the key file in a smart card if you're not sure of everyone in your own workplace (download the sidebar, "Achieve Security Nirvana With Key Files and Smart Cards" on the VSM Web site; see the Go Online box for details).

Now create a DLL. Open Visual Studio

.NET and create a Visual Basic Class Library project named "bizobj":

```
Public Class MatOper
    Public Shared Function Add(ByVal N1 As Double, ByVal N2 As Double) As Double
        Return N1 + N2
    End Function
    Public Shared Function Multiply(ByVal N1 As Double, ByVal N2 As Double) As Double
        Return N1 * N2
    End Function
End Class
```

This simple DLL adds and multiplies two numbers. In practice, you'll use a more complex piece of code that might include controls and forms. Now give the assembly a strong name. Change the AssemblyInfo.vb file in the project with this code:

```
<Assembly: AssemblyVersion("1.0.0.1")>
<Assembly: AssemblyKeyFile _>
    ("../../../../MyKey.snk")>
```

Visual Studio .NET has already created an "AssemblyVersion("1.0.*")" entry. You must replace that entry because it contains a "*" that Visual Studio interprets as asking it to increment the version number automatically. You don't want to get a new version at each compilation because it would produce a new strong name, which would force you to update all references to the file. Instead, change the version number manually when you do develop a version with new functionality. Don't let it happen automatically at each compilation when you're in the middle of the development process.

Copy the key file you created earlier to the project directory and compile it. After you generate the DLL, you can open it with the IL Disassembler (ILDASM.exe) and check the manifest file for the version number and presence of the public key. You can create as many DLLs as necessary at this point. These DLLs can instantiate WinForms or be Windows control DLLs themselves (download the bonus project on the Web site).

Now it's time to build the app. Create "Windows app" and add references to the DLLs created previously by selecting Project | Add Reference. Give the executable a strong name by changing the AssemblyInfo.vb file as before. Compile the app. You can open it with ILDASM and check its manifest (see Figure 1). Check that the reference to the business DLL includes its name, version, and public key token. Visual Studio .NET sometimes fancies "0.0.0.0" as the version number regardless of what you tell it. If so, change the version number, recompile, and change back again.

Learn Public Keys' Nicknames

The public key token provides a 64-bit hash of the public key that uniquely identifies the DLL's publisher. Think of the public key token as a nickname for your public key at a more manageable

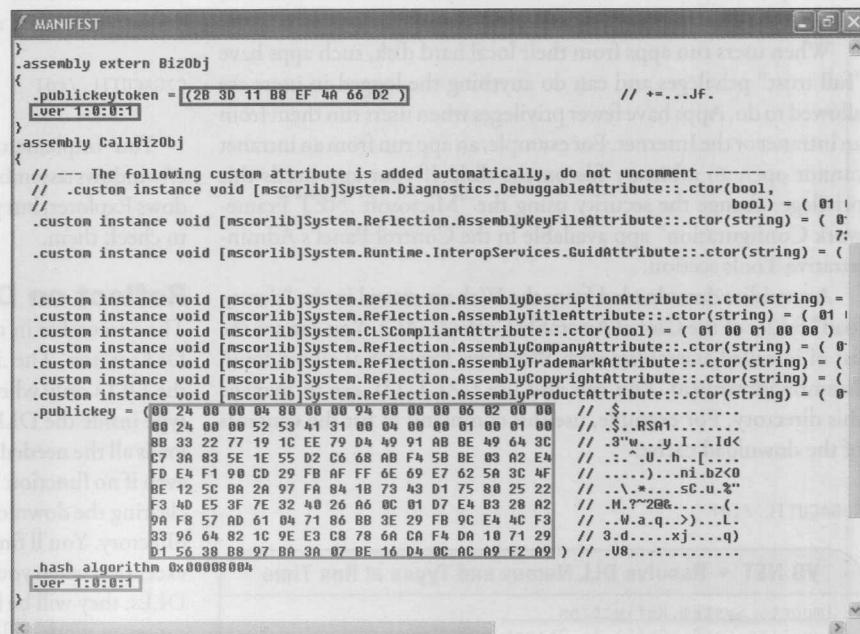


Figure 1 Deploy Win Apps on the Web Securely. Give your users the rich experience of Windows apps and the ease of deployment provided by the Web. Of course, Web deployment raises security issues, but you can handle that by using digital signatures to employ private and public keys, both for your EXE and associated DLLs. Open the digital signatures with ILDASM.exe. Here ILDASM reveals the MANIFEST section of a DLL, showing the DLL's own public key and version (red) and the referenced DLL version and public key token (blue).

length of eight bytes. Despite being much smaller, it will be different for each public key generated, according to the cryptographers at Microsoft. Use this command to obtain an assembly's public key token:

```
sn -T BizObj.dll
```

Users can run your app locally, but of course, your main objective is deploying it through HTTP. Create a directory in your Web server (usually at c:\inetpub\wwwroot), and call the directory WebDeploy. Copy both the EXE and the DLL there, and create a Web page as well. Call it Default.htm with a link to the executable:

```
<html><body>
<a href="CallBizObj.exe">Call executable</a>
</body></html>
```

Users can start the app in several ways. They can open Internet Explorer, type the URL http://localhost/webdeploy, and click on the link to execute the program. They also can use the Start command at the command prompt to say "Start http://localhost/webdeploy/callbizobj.exe." They can also create a shortcut on their desktop to http://localhost/webdeploy/callbizobj.exe. This way, the app behaves more like a non-Web-deployed Windows app and won't require users to open their browsers.

Users need to turn off any download managers they have (such as GetRight or GoZilla) before clicking on the link. Depending on their operating system, file locations, and security options, a message box might open and ask what to do with the executable. The users need to select Open if this happens. If that doesn't work, they

must adjust their .NET Framework security (download the sidebar, "Enable the Execution of Signed .NET Apps").

When users run apps from their local hard disk, such apps have "full trust" privileges and can do anything the logged-in users are allowed to do. Apps have fewer privileges when users run them from an intranet or the Internet. For example, an app run from an intranet cannot open an arbitrary file on the disk. If your app needs this privilege, change the security using the "Microsoft .NET Framework Configuration" app available in the Control Panel's Administrative Tools section.

Assemblies downloaded from the Web are stored in the "download" folder of the Global Assembly Cache (GAC). You can see the list of installed assemblies with Windows Explorer at c:\windows\assembly\download. You can also use GACUTIL.exe to manage this directory. For example, use this command to list the contents of the download cache:

```
C>GACUTIL /ldl
```

VB.NET • Resolve DLL Names and Types at Run Time

```
Imports System.Reflection
Function GetBizObj() As Object
    ' Declare variable to hold the assembly
    Dim MyAsm As [Assembly]
    ' Should use LoadFrom?
    If CheckBox1.Checked Then
        ' Load the DLL from the full URL
        MyAsm = [Assembly].LoadFrom(TextBox3.Text)
    Else
        ' Define the strong name
        Dim Name As New AssemblyName()
        Name.Name = "BizObj"
        Name.Version = New Version(1, 0, 0, 1)
        ' Your public key token will be different
        Name.SetPublicKeyToken(New Byte() {&H2B, &H3D, _
            &H11, &HB8, &HEF, &H4A, &H66, &HD2})
        Name.CultureInfo = Nothing
        ' Load the DLL
        MyAsm = [Assembly].Load(Name)
    End If
    ' Create an instance of the business object
    Return MyAsm.CreateInstance("BizObj.MatOper")
End Function
' Sum
Private Sub Button1_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) _
    Handles Button1.Click
    Dim N1 As Double = TextBox1.Text
    Dim N2 As Double = TextBox2.Text
    Dim R As Double = GetBizObj().Add(N1, N2)
    ListBox1.Items.Add(R)
End Sub
' Multiplication
Private Sub Button2_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) _
    Handles Button2.Click
    Dim N1 As Double = TextBox1.Text
    Dim N2 As Double = TextBox2.Text
    Dim R As Double = GetBizObj().Multiply(N1, N2)
    ListBox1.Items.Add(R)
End Sub
```

Listing 1 For runtime resolution, use reflection to load assemblies dynamically and instantiate types. Use the Assembly class method Load whenever possible. This program creates the strong name at run time, instantiates a class, then calls its methods. By resolving all the names at run time, you can express the program's structure in an external file (see Figure 1).

You can delete the contents of the download cache—useful during development—by typing this command:

```
C>GACUTIL /cdl
```

You implement the GAC as a series of directories under c:\windows\assembly. You can't examine the directories with Windows Explorer, but you can use the command prompt as a back door to check them.

Reflect on DLL Names and Types

The executable in my example contains information about all the DLLs it uses. The .NET Framework loader is kind enough to load the DLLs only when they're actually needed, such as when I use a type inside the DLL. This improves on Win32's behavior. Win32 loads all the needed DLLs at once when the program is first loaded, even if no function inside is needed. You can check this behavior by clearing the download cache, then deleting the DLL from the Web directory. You'll find that the program still executes, but it issues an exception when you press one of the buttons. An app can use several DLLs; they will be loaded on demand and stored in the GAC. You can even work offline (try it—select Work Offline from the File menu in Internet Explorer).

Sometimes you need to resolve the DLLs' names and types at run time. On such occasions, use reflection to load the assemblies dynamically and instantiate the types. Use either of two Assembly class methods: Load or LoadFrom. I usually choose Load because it's faster (see Listing 1). Create a strong name based on the DLL name, version, culture, and public key token, then load and instantiate it:

```
MyAsm = [Assembly].Load(Name)
MyAsm.CreateInstance("BizObj.MatOper")
```

Now change the HTML file Default.htm to link to this new executable:

```
<p><a href="CallBizObjRef.exe">Call BizObj using Reflection</a></p>
```

The apps developed in this article run inside a window separated from the browser. However, you can whip out a Windows control that runs seamlessly inside the browser window (see Additional Resources). You can also instantiate forms from apps started from the browser. And don't forget that your app can access data on the server through Web services. **VSM**

Mauro Sant'Anna is an MSDN regional director, a consultant, and a trainer with 20 years of development experience. He has been using the .NET Framework and Visual Studio actively since July 2000. Reach him at maurosantanna@hotmail.com.

Additional Resources

- *Applied Microsoft .NET Framework Programming* by Jeffrey Richter [Microsoft Press, 2002, ISBN: 0735614229]
- "INFO: How Internet Explorer Determines Positions for .NET Framework Assemblies": <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q311301>
- "Using Windows Forms Controls in Internet Explorer": www.gotdotnet.com/team/windowsforms/iesourcing.aspx



Relational
database

Object-oriented
development

If your back-end database isn't a good match for your front-end development, you need a new database. Caché, the high-performance database from InterSystems, is a powerful fusion of today's mainstream technologies: objects and SQL.

Unlike Oracle and other relational databases, Caché takes advantage of its efficient multidimensional data engine to implement an advanced object model. It doesn't try to hide a cumbersome relational engine beneath object-like wrappers.

Every Caché object is compatible with Java, C++, ActiveX, and other rapid development technologies. And thanks to Caché's "Unified Data Architecture," every object class is instantly accessible as tables via ODBC and JDBC.

With no mapping or middleware. That means no wasted development time. And no extra processing at run time. So not only will your applications be quick to build or adapt, they will run faster too.

Time to Change Your Database.



Make Applications Faster

Download Caché for free or request it on CD at www.InterSystems.com/match17

Keep a Window Visible

Prevent users from moving windows offscreen, work with unsafe and unchecked code, and make a field variable declaration behave like a constant.

by Karl E. Peterson, Juval Löwy,
and Mattias Sjögren

Technology Toolbox

- VB.NET
- C#
- SQL Server 2000
- ASP.NET
- XML
- VB6
- Note:

Karl E. Peterson's solution
also works with VB5.

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these resources.

Download

VS0209QA Download the code for this article, which includes VB6 code demonstrating how to keep a form fully onscreen, and VB.NET and C# code showing how to use read-only variables.

Discuss

VS0209QA_D Discuss this article in the C# forum.

Read More

VB0209QA_T Read this article online.

VB0102JF_T "Drill Down on VB.NET" by James Foxall

Q: Keep Your Window Visible

How can I prevent a user from moving a window off the screen? I'd like to allow users to move a window around the desktop at will, but not allow them to move any part of the window off the desktop. The entire window must be viewable at all times.

A:

Hopefully, this is a client specification and not a method to keep some sort of popup advertisement perpetually in someone's face. (Online, a <g> would probably follow, but irritating users is rarely funny.) Assuming a legitimate need, the simple answer is that you'd want to hook your form's message stream and respond to incoming WM_MOVING messages (see Additional Resources). If you don't have a favorite drop-in subclassing module, I'd urge you to grab HookMe.zip from my Web site or this column's sample code (download the code from the VSM Web site; see the Go Online box for details).

Windows sends WM_MOVING messages to a window immediately prior to the user getting any feedback. These messages are accompanied by a pointer to a RECT structure in lParam that contains the drag rectangle coordinates Windows displays to the user. You're only given a pointer, so you need to copy the data at this address to a RECT structure declared within your hook procedure (see Listing 1).

At this point, you're free to examine the RECT coordinates and even modify them to suit your needs. In this case, you'd want to en-

sure that none of the edges go past the edge of the screen, and if they do, correct them to remain onscreen. After any necessary modifications, copy the updated structure back to the same address passed in lParam and tell Windows you've handled the message by returning True for the function result. —K.E.P.

Q: Work With Unsafe Code

I hear you can do pointers in C#, and that you can manipulate memory that way. How is it done? Is the memory still managed by .NET? Is the code still managed code?

A:

C# does support direct memory manipulation using pointers. Such C# code is called unsafe code, because this code lets go of most of the safety of .NET memory management. However, unsafe code is still managed code, because it runs in the Common Language Runtime (CLR), and .NET still garbage collects it. C# supports unsafe code to ease the task of porting legacy C++ applications to C# in cases that use

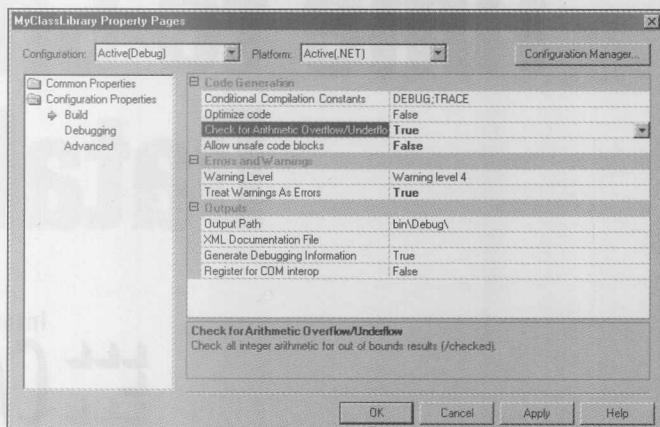


Figure 1 Configure Project Settings. The Properties page lets you configure project settings, including warning level, unsafe code support, unchecked code support, and treating warnings as errors. You can configure different settings for Debug and Release builds.

```
Option Explicit
```

```
Private Declare Sub CopyMemory Lib "kernel32" _
    Alias "RtlMoveMemory" (Destination As Any, _
    Source As Any, ByVal Length As Long)

Private Const WM_MOVING As Long = &H216

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

' ****
' Subclassing
' ****

Friend Function WindowProc(hWnd As Long, _
    Msg As Long, wParam As Long, lParam As Long) _ As Long
    Dim Result As Long
    Dim r As RECT
    Dim dX As Long
    Dim dY As Long

    ' Precalculate screen dimensions.
    dX = Screen.Width \ Screen.TwipsPerPixelX
    dY = Screen.Height \ Screen.TwipsPerPixelY

    Select Case Msg
        Case WM_MOVING
            ' Grab screen coordinates of drag rectangle.
```

```
Call CopyMemory(r, ByVal lParam, _
    Len(r))
    ' Adjust to prevent window from going offscreen.
    If r.Left < 0 Then
        r.Right = r.Right - r.Left
        r.Left = 0
    End If
    If r.Top < 0 Then
        r.Bottom = r.Bottom - r.Top
        r.Top = 0
    End If
    If r.Right > dX Then
        r.Left = dX - (r.Right - r.Left)
        r.Right = dX
    End If
    If r.Bottom > dY Then
        r.Top = dY - (r.Bottom - r.Top)
        r.Bottom = dY
    End If
    ' Update drag rectangle for Windows.
    Call CopyMemory(ByVal lParam, r, Len(r))
    ' Let Windows know we've handled this.
    Result = True
Case Else
    ' Pass along to default window procedure.
    Result = InvokeWindowProc(hWnd, Msg, _
        wParam, lParam)
End Select

' Return desired result code to Windows.
WindowProc = Result
End Function
```

Listing 1 You can alter the position displayed as the user drags a window about the screen. Intercept the WM_MOVING message and alter the contents of the rectangle structure used by Windows to position the window's drag rectangle.

complex pointer arithmetic. This is probably why VB.NET doesn't support unsafe code.

Unsafe code also comes in handy when interoperating with Win32 API calls that require pointers. C# unsafe code uses C-like pointer syntax for the most part. You can only use unsafe code at a method's scope by prefixing the method definition with the reserved word "unsafe," then using C-pointer syntax for direct memory manipulation:

```
unsafe public void MyUnsafeMethod1(int* ptr)
{
    *ptr = 1;
    *(ptr + 1) = 2;
    *(ptr + 2) = 3;
    Debug.Assert(ptr != null);
    *ptr = 3;
}
```

You can only apply the unsafe qualifier to methods and properties, not to individual statements or class member variables. Note one important programming detail when dealing with unsafe code: You must pin down the memory sections you interact with directly, using a fixed statement, because

garbage collection can start at any moment and move objects around in memory. The fixed statement takes this form:

```
fixed(type* ptr = expr)
```

It pins down the object ptr points at, while the expression in the statement executes:

```
unsafe void UnsafeArrayAccess()
{
    int[] intArray = new int[3];
    fixed(int* ptr = intArray)
    {
        *ptr = 1;
        *(ptr + 1) = 2;
        *(ptr + 2) = 3;
        Debug.Assert(ptr != null);
        *ptr = 3;
    }
}
```

You don't need to pin down unsafe access to value types because value types are stack allocated, so they aren't subjected to garbage collection:

```
struct Point
```

```

public int x;
public int y;
}
//using unsafe struct
unsafe void UnsafeStructUsage()
{
    Point point;
    point.x = 1;
    point.y = 2;

    Point* pPoint = &point;
    pPoint->x = 3;
    pPoint->y = 4;
}
```

The C# compiler doesn't support unsafe code by default—you must enable it explicitly in your project configuration. In the Project Properties page, select Build, and set the "Allow unsafe code blocks" dropdown box to True (see Figure 1). —J.L.

Q: Understand Checked and Unchecked Code

What is checked code? How is it different from normal C# code? Is it the same as managed code?

A:

By default, the C# compiler and the CLR don't check for overflow or underflow after performing arithmetic operations. This is called unchecked code. As a result, you might get erroneous results without knowing it, even though it's valid managed code. For example, consider the CalcPower() method that returns the result of a specified number raised to a specified power:

```
int CalcPower (int num,int power)
{
    int result = 1;
    for(int i = 1;i<=power;i++)
    {
        result *= num;
    }
    return result;
}
```

Because int is only 32 bits long, trying to calculate CalcPower(10,11) returns the bogus result of 1,215,752,192 instead of 100,000,000,000. You can instruct the C# compiler to throw an exception of type OverflowException in case of an overflow error, using the checked instruction:

```
int CalcPower(int num,int power)
{
    int result = 1;
```

Add unique PDF features to all your applications



AMYUNI PDF CONVERTER

Convert any Windows® document to PDF* format

Concatenate and merge multiple documents into one PDF file

Add bookmarks and hyperlinks to the resulting PDF document

Interface with most Windows® programming languages

Available versions

- Single-user or server license
- Royalty-free Developer license
- Unlimited Site license
- Open source-code license
- OEM License

AMYUNI Consultants Contacts

Info: sales@amyuni.com

Evaluation: www.amyuni.com

EUROPE

Sales: (+33) 1 30 61 07 97

AMERICAS

Sales: (703) 937 3619

AMYUNI PDF CREATOR

View, Edit and Print PDF* documents

Create complex documents, forms and reports linked to your database

Comes as an ActiveX component for easy integration into your application

* Portable Document Format

Visit us also at:
www.amyuni.co.uk
www.amyuni.fr

www.amyuni.com

All trademarks are property of their respective owners. © 2001 AMYUNI Consultants, all rights reserved.

```
for(int i = 1;i<=power;i++)
```

```
{ checked
```

```
{ result *= num;
```

```
}
```

```
}
```

```
return result;
```

Now, the calling client is aware that an error took place. Similarly, you can flag a code segment as explicitly unchecked using the unchecked instruction:

```
unchecked
```

```
{ //some code }
```

```
}
```

You can nest checked or unchecked statements inside each other. By default, the C# compiler generates unchecked code. To enable support for checked code, open the Project Properties page, select Build, and set "Check for Arithmetic Overflow/Underflow" to True (see Figure 1). You should use checked code for the "usual suspects"—that is, calculating powers, calculating factorials, and so on. —J.L.

Q: Declare Constant Object Variables

I wrote a class and now I want to declare a constant instance of that class, but the VB.NET compiler won't let me. Is there a workaround?

A:

The only types you're allowed to use in a VB.NET Const statement are the ones you can write literals for. These are the primitive types (numeric types, Boolean, Char, Date), String, and Object. The only value you can initialize an Object constant to is Nothing, because no allocation can occur.

The same rules apply to C#, except it doesn't provide a way to write date literals, so it doesn't support DateTime constants. On the other hand, it allows you to declare a null constant of any reference type, not only Object.

Fortunately, VB.NET has another modifier keyword, ReadOnly, which you can apply to a field variable declaration to make it behave almost like a constant. You can initialize a read-only variable where it's declared or in a constructor, but you're not allowed to change it after that. The compiler enforces this rule, so you get a compile error if you try to modify a variable marked as ReadOnly in a method.

Take a look at a couple different ReadOnly variables in use (see Listing 2). The code illustrates an important difference compared to constants. When you use a read-only variable, you can change how it's initialized based on calculations and parameters passed to the constructor. You're not limited to a single static value the way you are when declaring a constant.

Another significant distinction between the two is related to versioning. When you use a constant, its value is embedded in the code everywhere it's used. The CLR never refers to the constant field at run time, and in fact doesn't even load it into memory. This wouldn't work for a read-only field, because it's only semi-constant, and the VB.NET compiler can't determine its value at compile time.

48

VISUAL STUDIO MAGAZINE • SEPTEMBER 2002 • www.visualstudiomagazine.com

VB.NET • Compare Const With ReadOnly

Class Ball

```
Private Const PI As Double = 3.14159265
' The following doesn't work
' Public Const SmallBall As New Ball( 5 )

Public Shared ReadOnly SmallBall As New Ball( 5 )
Public Shared ReadOnly BigBall As New Ball( 20 )

Public ReadOnly Radius As Integer
Public ReadOnly Volume As Double

Public Sub New(ByVal radius As Integer)
    Me.Radius = radius
    Volume = 4 / 3 * PI * radius^3
End Sub

End Class
```

Listing 2 You use constants for only simple, static values. The `ReadOnly` keyword offers more flexibility because it allows you to initialize the variable in a constructor based on calculations and parameters.

Because of this, you should only use constants for values you're certain will never change. If you change a constant, you'd have to recompile all code that uses it for the change to take effect.

Also note that the VB.NET `Const` keyword implies `Shared`, meaning you can always access the constant without having to instantiate the class. `ReadOnly` doesn't do this, so if you want a read-only field to be `Shared`, you must add the `Shared` keyword explicitly.—M.S.

Karl E. Peterson is a GIS analyst with a regional transportation planning agency and serves as a member of the *Visual Studio Magazine* Technical Review and Editorial Advisory Boards. Online, he's a Microsoft MVP and a section leader on several DevX forums. Find more of Karl's VB samples at www.mvps.org/vb.

Juval Löwy is the principal of iDesign, a consulting and training company focused on .NET design and migration. Juval is the author of *COM and .NET Component Services* (O'Reilly), and he speaks frequently at major conferences. Juval chairs the program committee of the .NET California Bay Area User Group. Contact him at www.idesign.net.



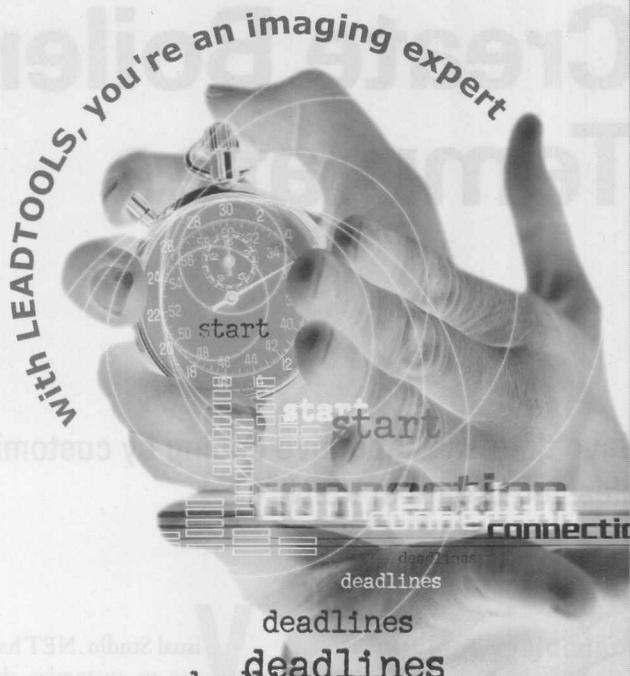
Mattias Sjögren lives in southern Sweden, where he tries to combine consulting work with university studies. He is a Microsoft MVP for Visual Basic. Reach him at mattias@mvps.org or visit his Web site at www.msjogren.net.

Additional Resources

- [HookMe.zip](http://www.mvps.org/vb/samples.htm): www.mvps.org/vb/samples.htm
- "Read-Only Variables": http://msdn.microsoft.com/library/en-us/vbls7/html/vblrfvbspec7_5_2.asp
- "App Object Changes in Visual Basic .NET": <http://msdn.microsoft.com/library/en-us/vbcon/html/vxconchangepstoappobjectinvisualbasicnet.asp>

LEADTOOLS®

LEADING technology in imaging developer toolkits



For the past 12 years, programmers worldwide have recognized LEAD Technologies as the undisputed LEADER in imaging developer toolkits. LEADTOOLS handles all of your imaging needs, from common loading, displaying and image processing, to the complex and high performance imaging demands of the document, medical and Internet industries. Whether your project requires raster imaging, vector imaging or multimedia support, (or all of the above!) LEADTOOLS is your solution. Why struggle with multiple products from multiple vendors when you can get all of your imaging development needs in one toolkit? One toolkit...

...now you're an imaging expert!

leadtools.com

.NET API C++ Lib. COM ActiveX VCL

There are versions of LEADTOOLS to fit any digital imaging project including **raster**, **multimedia**, **document**, **medical**, **Internet** and **vector**.



**LEAD
TECHNOLOGIES**

sales@leadtools.com or call: 800-637-4699

1201 Greenwood Cliff, Suite 400 Charlotte, NC 28204

LEAD and LEADTOOLS are registered trademarks of LEAD Technologies, Inc.

Create Boilerplate Templates

Save time and repetitive coding by customizing the code contained in new project files you create in VS.NET.

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209AN Download the code listings for this article separately.

Discuss

VS0209AN_D Discuss this article in the ASP.NET forum.

Read More

VS0209AN_T Read this article online.

N020430AN "Page Templates Revised" by Jonathan Goodyear

VS0112AN_T ASP.NET, "Create Page Templates," by Jonathan Goodyear

Visual Studio .NET has many hooks you can use to customize the way it behaves. If you're frustrated by something—such as the limited boilerplate code VS.NET's New Project Item Wizard produces (see Listing 1)—there's usually a way to customize it to suit your needs. In this column, I'll show you how to customize the code VS.NET creates for files such as class libraries and ASP.NET Web Forms by stepping you through the process of creating your own project item template. You'll save yourself time, trouble, and repetitive coding, and give your future .NET projects increased standards compliance.

Each of the New Project Item Wizards has a template contained under the VS.NET file path. I'll assume you didn't modify your install path when you installed VS.NET, and that it's located at C:\Program Files\Microsoft Visual Studio .NET. I'll call this folder the root folder. The root folder contains a subfolder for each supported .NET language that's installed. I'll show you how to create a C# template in this article, but you can mirror the instructions to create a Visual Basic .NET template as well.

The VC# subfolder under

the root folder contains two subdirectories you'll need for this project: CSharpProjectItems and VC#Wizards. The CSharpProjectItems subfolder contains a LocalProjectItems subfolder and a WebProjectItems subfolder. Expand the WebProjectItems subfolder because you'll be building a Web application. Notice that this folder and all its subfolders match up exactly with VS.NET's Add New Item Wizard. Create

ASP.NET, C# • Create Your Own Wizard Templates

```
<%@ Page language="c#" %>
Codebehind="WebForm3.aspx.cs"
AutoEventWireup="false"
Inherits="TestingStuff.WebForm3" %>
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
<head>
<title>WebForm3</title>
<meta name="GENERATOR" Content=
"Microsoft Visual Studio 7.0">
<meta name="CODE_LANGUAGE" Content="C#"/>
<meta name="vs_defaultClientScript"
content="JavaScript">
<meta name="vs_targetSchema" content=
"http://schemas.microsoft.com/intellisense/ie5">
</head>
<body MS_POSITIONING="GridLayout">
<form id="WebForm3" method="post"
runat="server">
</form>
</body>
</html>
```

Listing 1 The wizard templates that come with VS.NET are rudimentary because they need to be applicable to all development projects. Create your own wizard templates and add the boilerplate code that's specific to your company and its projects.

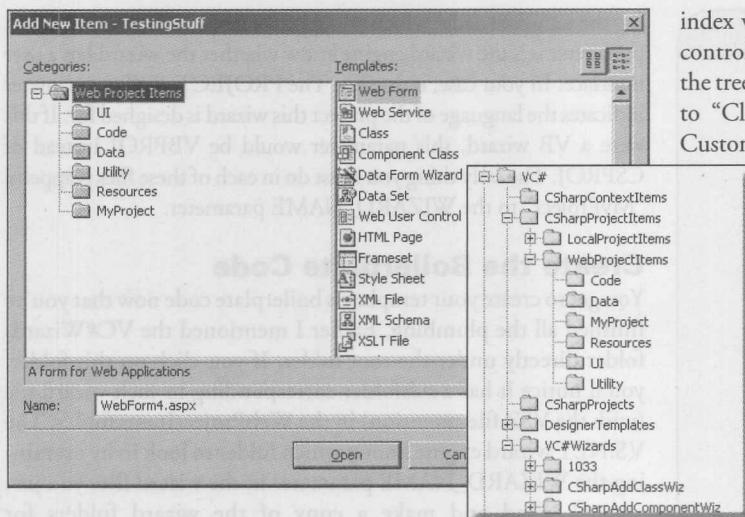


Figure 1 Add Your Own Folder. VS.NET's Add New Item Wizard follows the same structure as the VS.NET folders in the file system. If you add your own folder to hold your custom wizard templates, it shows up in the wizard. The MyProject folder shown in the wizard holds the custom wizard templates created in this article.

a new folder named "MyProject"; it's added to the wizard (see Figure 1). However, if you click on the MyProject folder in the wizard, you'll see that it's empty. To add your own templates, you need to add a file to your MyProject folder named MyProject.vsdir. This is a VS.NET directory file, and you can use it to point VS.NET to the wizards to build your file templates. You can expedite the process by copying the WebProjectItems.vsdir file from the WebProjectItems folder and renaming it.

Open the MyProject.vsdir file in Notepad; it contains a cryptic-looking listing for each type of file you can create using the Add New Item Wizard. Make things simple for now, and remove all the file types except for the ones that begin with "..\CSharpAdd-WebFormWiz.vsz" and "..\CSharpAddClassWiz.vsz," including the lines that start with UI, Code, Data, Utility, and Resources. Your file should now have only two rows of data in it. It's never a good idea to overwrite any of the existing templates, so append "..\MyProject" to the beginning of each of these wizard filenames, so that now the rows begin with "..\..\MyProjectCSharpAddWebFormWiz.vsz" and "..\..\MyProjectCSharpAddClassWiz.vsz." You need the extra "..\" because your folder is one level deeper in the directory structure, and these are paths to actual wizard files (which I'll talk about a bit later).

Be Careful What You Change

At this point, you might be tempted to fiddle with some of the other pieces of data for each row in the file. Be careful when you do this. The Globally Unique Identifier (GUID) you see on each line is the GUID for C# itself and must remain there. The numbers with pound signs (#) next to them are indexes into a private resource file that determines the name and description of the item that's displayed in the Add New Item Wizard.

You can change your custom project item's name by substituting the first index value with a textual name. For now, change the first index values of your custom project items to MyProjectClass and MyProjectWebForm. Be sure to remove the pound sign from the

index value as well. The second index value with the pound sign controls the extended description that appears in the gray box below the treeview in the Add New Item Wizard. Change the index values to "Class library Customized for MyProject" and "Web Form Customized for MyProject."

The fourth item in the bar (l) delimited list of data is a number that determines the order in which the items are displayed in the wizard. The item with the lowest number is positioned at the top, and so on. Set these ordering index values to 20 and 10, which makes MyWebForm appear first in the Add New Item Wizard. The only other piece of data that makes sense to customize is the last item on each row. Each of the files created by your template gets this default name unless you change it in the Add New Item Wizard (numbers are appended to the end to prevent duplicate filenames). To demonstrate, change the Web Form default name to Screen.aspx, and the class

library default name to LogicClass.cs.

Earlier, you changed the names of the wizard file pointers in your

ASP.NET, C# • Expand on VS.NET's Web Form Wizard Template

```
<%@ Page language="c#" Codebehind="$FILENAME$.cs"
AutoEventWireup="false" Inherits="$INHERITS$"
%>
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.0 Transitional//EN" >

<html>
  <head>
    <title>XYZ Company - MyProject: [<output
      SAFE_ITEM_NAME]</title>
    <link rel="stylesheet" href=
      "stylesheets/xyz_intranet.css"
      type="text/css">
    <meta name="GENERATOR" Content=
      "Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" Content="C#"/>
    <meta name=vs_defaultClientScript content=
      "[<output DEFAULT_CLIENT_SCRIPT]">
    <meta name=vs_targetSchema content=
      "[<output DEFAULT_TARGET_SCHEMA]">
  </head>
  <body MS_POSITIONING=
    "[<output DEFAULT_HTML_LAYOUT]">

    <form id=" [<output SAFE_ITEM_NAME]"
      method="post" runat="server">
      <h1>Put title here</h1>
      <br>
    </form>

    <center>
      <b>Copyright 2002 XYZ Company - All rights
      reserved.</b>
    </center>
  </body>
</html>
```

Listing 2 This custom wizard template expands upon the existing Web Form wizard template that comes with VS.NET. It adds the company name and project name to the page title, a placeholder for the page name, and a copyright notice at the bottom of the page. In your own templates, you can add as much code as you need to reduce your workload and add consistency to your pages.

C# • Implement Exception Handling

```

using System;
using System.Data;
using System.Data.SqlClient;

namespace [!output SAFE_NAMESPACE_NAME]
{
    /// <summary>
    /// XYZ Company - MyProject
    /// This code is for internal purposes only.
    /// [&!output SAFE_CLASS_NAME]
    /// </summary>
    public class [&!output SAFE_CLASS_NAME]
    {
        public [&!output SAFE_CLASS_NAME]()
        {
            try
            {
                // TODO: Add constructor logic here
            }
            catch(Exception ex)
            {

            }
            finally
            {
            }
        }
    }
}

```

Listing 3 All projects should implement exception handling, so you might as well create a custom wizard template that implements it. Company descriptions, legal notices, and detailed comment placeholders are some other things you might want to consider for your own class library templates.

MyProject.vsdir file. Notice how each wizard file pointer begins with "..\\"? Well, traverse two levels up the directory tree to the CSharpProjectItems folder, and you'll find a file with a VSZ file extension for each template in the Add New Item Wizard. Make a copy of the CSharpAddWebFormWiz.vsz file and rename it to MyProjectCSharpAddWebFormWiz.vsz. Do the same thing for the CSharpAddClassWiz.vsz file. Inside each of these files is the information necessary to launch your template wizard:

```

VSWIZARD 7.0
Wizard=VsWizard.VsWizardEngine
Param="WIZARD_NAME ="
    "MyProjectCSharpAddWebFormWiz"
Param="WIZARD_UI = FALSE"
Param="PROJECT_TYPE = CSPROJ"

```

The first line indicates that this wizard is meant to run using VS7 (numbered for compatibility with future versions of VS.NET). The Wizard item indicates the wizard engine to use. In the three rows that begin with "Param=", the WIZARD_NAME parameter is the name of the wizard folder contain-

ing the template code (which you'll create next). The WIZARD_UI parameter lets the wizard engine know whether the wizard has a user interface. In your case, it doesn't. The PROJECT_TYPE parameter indicates the language of the project this wizard is designed for. If this were a VB wizard, this parameter would be VBPProj instead of CSPROJ. The only thing you must do in each of these files is append "MyProject" to the WIZARD_NAME parameter.

Create the Boilerplate Code

You get to create your template's boilerplate code now that you've finished all the plumbing. Earlier I mentioned the VC#Wizards folder directly under the root folder. If you click on this folder, you'll notice it has a subfolder corresponding to each wizard file (with the VSZ file extension) in the WebProjectItems folder. The VS.NET wizard engine knows which folder to look in by examining the WIZARD_NAME parameter in the wizard files you just created. Find and make a copy of the wizard folders for CSharpAddWebFormWiz and CSharpAddClassWiz, appending "MyProject" to each one.

Each of your wizard folders has two subfolders, Scripts and Templates. The Scripts folder contains the default.js file (under another subfolder named "1033"). Don't modify this file, because it uses symbols that are private to VS.NET. These symbols can change at any time, so you risk breaking your template if you change anything in that file. Forget it exists. The Templates folder (actually, the "1033" subfolder underneath it) contains the files you can modify. The MyProjectCSharpAddWebFormWiz wizard presents you with two files. The Templates.inf file merely points to the files needed for the wizard (only one in the case of both your new wizards). You don't need to rename the file listed in Templates.inf, because VS.NET extracts the proper name for your new files from the MyProject.vsdir file you built earlier.

Open the WebForm1.aspx file and add any custom code you want. You'll notice a few symbols such as \$FILENAME\$, SAFE_ITEM_NAME, DEFAULT_CLIENT_SCRIPT, and DEFAULT_HTML_LAYOUT. You can change or remove these, but I wouldn't recommend it. VS.NET uses them to wire up the templates to the

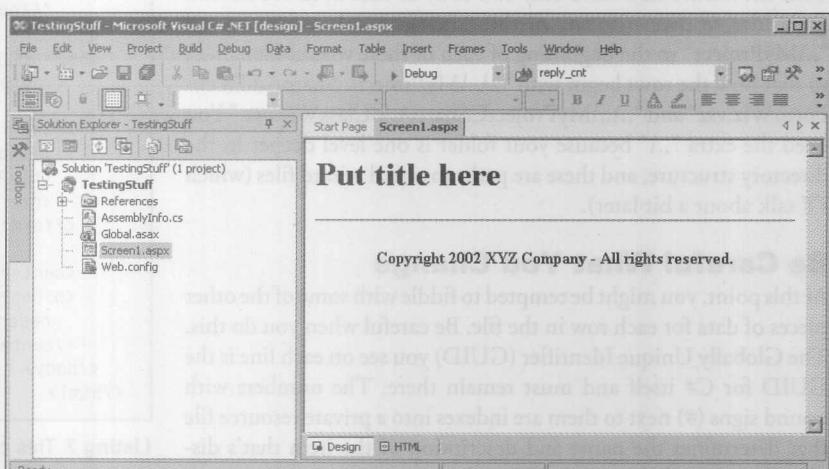


Figure 2 Add Boilerplate UI Code. Instead of being greeted with an empty screen when you create a new Web Form, you can create your own custom wizard templates to inject boilerplate user interface code that needs to exist on every page. You can also add placeholders for items you don't want to forget while developing your Web Forms.

file/class names that you specify in the Add New Item Wizard, as well as your VS.NET IDE preferences.

As an example, I customized the <title> HTML tag with the name of a fictional business (XYZ Company) and the project name (MyProject). I also added a default page title and <hr> tag, as well as a copyright message at the bottom (see Listing 2). In a real-life project, you'd want to add any code that every page needs to implement but might be forgotten otherwise. Open up the NewCSharpFile.cs file in the "Templates\1033" subfolder of the MyProjectCSharpAddClassWiz wizard folder. You might want to replace the SAFE_NAMESPACE_NAME symbol with the namespace you want each of your classes to use, but you should leave the SAFE_CLASS_NAME symbol alone, because it corresponds to the filename you specify in the Add New Item Wizard. Some appropriate things to add to class templates are commonly used namespace references, default comments (including company and project name), licensing disclaimers, and default exception-handling logic (see Listing 3).

Now that you've finished your wizard template, try it out in VS.NET. If you have VS.NET open already, shut it down and restart it. The IDE caches your previous wizard settings, so you want to force them to be reloaded with your new wizard options. Open a C# ASP.NET project, and open the Add New Item Wizard. Your MyProject folder should be there. Click on it and select the Web Form option. Notice that it indicates Screen1.aspx as the default

filename. Click on the Open button. You should be presented with a Web Form that reflects the boilerplate code you added to your wizard template (see Figure 2).

The steps you've taken to create your wizard template might seem like a lot of trouble to go through, but remember all the time having a boilerplate template will save you. I hope I've also encouraged you to explore VS.NET to find some of the neat (and often undocumented) things you can do with it. Remember to always make backup copies of any configuration files before you modify them, in case something goes wrong and you need to revert back to the way things were before. **VSM**

Jonathan Goodear is the president of ASPSoft (www.aspsoft.com), an Internet consulting firm based in Orlando, Fla. He's MSDC-, MCP-, and CLS-certified and is the coauthor of *Debugging ASP.NET* (New Riders Publishing). Reach him at jon@aspsoft.com or through his angryCoder eZine at www.angryCoder.com.



Debugging
ASP.NET

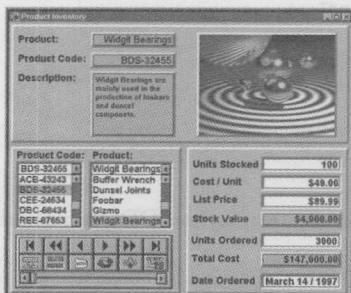
Jonathan Goodear
Orlando, Fla.

Additional Resources

"Understanding Templates in ASP.NET" by Dino Esposito [*MSDN Magazine* January 2002]: <http://msdn.microsoft.com/msdnmag/issues/02/01/cutting/cutting0201.asp>

The Fastest Database Engine

Discover **CodeBase**, the **fastest** database engine on the market that's also fully **xBASE file compatible**. Query a million-record table in 0.34 seconds, or read 300,000 records in just 0.59 seconds. It's lightning quick!



ADO/ODBC Support!
This application was created without a single line of source code! Use native and 3rd party controls with our new ADO/OLE-DB and ODBC support. Creating database programs has never been easier!

- ✓ Multi-user file compatible with FoxPro, Clipper and dBASE
- ✓ Client/Server, multi-user & single user support included
- ✓ Small DLL loads quickly, simply and use few resources
- ✓ 100% Portable: Supports all popular compilers, OS's and the internet
- ✓ Reader's Choice Award for 5 Years
- ✓ Royalty Free Distribution

FREE 30 Day Test Drive!

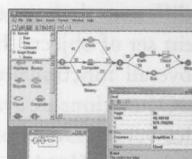
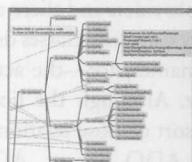
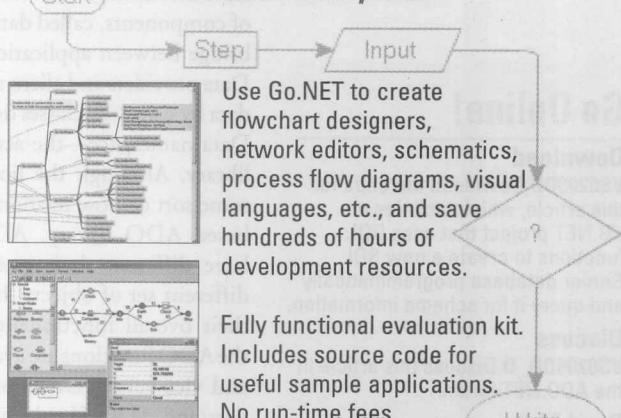
Phone: (780) 437-2410
Fax: (780) 436-2999

SEQUITER®
SOFTWARE INC.

www.codebase.com - info@codebase.com

BUILD INCREDIBLE INTERACTIVE GRAPHICS WITH GO.NET

Written entirely in C#



Use Go.NET to create flowchart designers, network editors, schematics, process flow diagrams, visual languages, etc., and save hundreds of hours of development resources.

Fully functional evaluation kit. Includes source code for useful sample applications. No run-time fees.

Only \$895.

Ask about our Autolayout package.

www.nwoods.com

Call now for special

introductory offer

800.434.9820

603.886.5780

Northwoods
SOFTWARE CORPORATION

Manipulate Schema Information

Explore your options for manipulating a table's schema programmatically.

by Dino Esposito

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**

ADO.NET doesn't provide classes for schema manipulation, although Microsoft plans to build an all-encompassing .NET object model for the data source's Data Definition Language (DDL). In this column, I'll examine the three options you have today to manipulate a table's schema programmatically—to create tables, views, indexes, constraints, and more. The ADO.NET options for accessing and modifying a data source's schema are the data source's SQL interface, the default programming interface of the data source's OLE DB provider, and the COM ADOX object model's functions—namely, the Microsoft ActiveX Data Objects Extensions for Data Definition Language and Security (ADOX).

Some differences between ADO and ADO.NET become apparent when you consider that in the .NET Framework, a new breed of components, called data providers, serve as a bridge between applications and data sources. Data providers and client applications exchange data by using the classes defined in the System.Data namespace—the actual ADO.NET class library. Although the library's name suggests some sort of close relationship with the COM-based ADO library, ADO and ADO.NET have different design centers and feature a different set of objects. In addition, although their overall functionalities intersect, not all the ADO functions are available in ADO.NET, and vice versa. (Download the sidebar, "Understand ADO Headaches in .NET Applications," from the VSM Web site; see the Go Online box for details.)

Any commercial DBMS provides you with a set of SQL commands or other programmable tools, which you can use to execute DDL statements. A DDL statement lets you perform schema manipulation on a database or a catalog. Typical DDL statements are CREATE TABLE and ALTER COLUMN, which respectively

create a new table and modify parameters on a column.

From a .NET data provider's perspective, a DDL statement is the same as a SQL command. Like INSERT or UPDATE statements, a DDL statement doesn't return rows, so you only need to use the Command object's ExecuteNonQuery method to process it. You can create a new database element in SQL Server's local installation:

```
Dim s As String
s = "SERVER=localhost;UID=sa;"
Dim conn As SqlConnection
conn = New SqlConnection(s)
```

```
Dim c As String
c = "CREATE DATABASE MyData"
Dim cmd As SqlCommand
cmd = New SqlCommand()
cmd.CommandText = c
cmd.Connection = conn
conn.Open()
cmd.ExecuteNonQuery()
conn.Close()
```

In particular, employ this DDL command:

```
CREATE DATABASE MyData
```

This command creates a new database with all the default settings for the size of the database's primary file and transaction log. Also, you don't specify a MAXSIZE, so the files can grow to fill all available disk space. You can create new databases with special characteristics programmatically by using a more articulated version of the CREATE DATABASE command. Similarly, you can create tables with the specified number of columns, primary keys, and constraints (see the T-SQL documentation in Books Online for how to accomplish this task with

Go Online!

Download

VS0209DD Download the code for this article, which includes a VB.NET project that uses DDL functions to create a new SQL Server database programmatically and query it for schema information.

Discuss

VS0209DD_D Discuss this article in the ADO.NET forum.

Read More

VS0209DD_T Read this article online. It includes the extra sidebar, "Understand ADO Headaches in .NET Applications."

VB9906JM_T "Ease Data Access With ADOX" by Jeffrey P. McManus

VS0111GS_T Getting Started, "Share Data Between Database Formats," by Andy Rosebrock

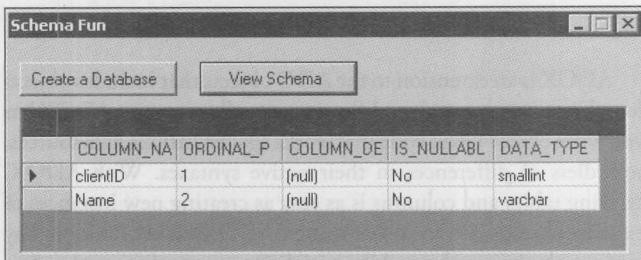


Figure 1 Integrate Information With the Rest of Your App. The sample application creates a new SQL Server database with one child table, then reads back the information about the newly created table. It also allows you to view the table schema using both SQL Server statements and the OLE DB functions.

SQL Server). This article's sample code uses this command to create a MyClients table with two columns (download the code from the VSM Web site). One column acts as the primary key and is autoincremented:

```
USE MyData
CREATE TABLE MyClients(
    clientID smallint IDENTITY(1, 1) PRIMARY KEY,
    Name varchar(80) NOT NULL)
```

The database DDL implementation also comes in handy if you need to read schema information from an existing database. For example, suppose you want to read back the structure of the just-created MyData database. Any SQL Server database maintains schema information in system tables accessible by INFORMATION_SCHEMA views. The INFORMATION_SCHEMA.COLUMNS view contains schema information about each table defined within the catalog. Available columns include column name, data type, maximum character number, whether the column is nullable, and more (see Listing 1 for how to set up this DDL command for execution):

```
USE MyData
SELECT COLUMN_NAME,
    ORDINAL_POSITION,
    COLUMN_DEFAULT,
    IS_NULLABLE,
    DATA_TYPE
FROM
    INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'MyClients'
```

ADO.NET returns the information to you as a DataSet object so you can integrate it easily with the rest of the application (see the results in Figure 1).

Use OLE DB Features

You can resort to the OLE DB programming interface to access schema information if you don't plan to use the DBMS.NET native data provider, or if a .NET provider isn't available yet for your database (if you're using Sybase, Informix, or DB2).

The OLE DB .NET data provider makes schema information available through a method on the OleDbConnection object called

VB.NET, SQL Server • Get a Particular SQL Server Catalog's Schema Information

```
Function GetSqlSchemaInfo() As DataSet
    Dim s As String
    s = ConnectionString
    Dim conn As SqlConnection
    conn = New SqlConnection(s)

    Dim c As String
    c = "SELECT "
    c += "COLUMN_NAME, "
    c += "ORDINAL_POSITION, "
    c += "COLUMN_DEFAULT, "
    c += "IS_NULLABLE, "
    c += "DATA_TYPE FROM "
    c += "MyData.INFORMATION_SCHEMA.COLUMNS "
    c += "WHERE TABLE_NAME = 'MyClients'"

    Dim cmd As SqlCommand
    cmd = New SqlCommand(c)
    cmd.Connection = conn
    cmd.CommandText = c

    Dim da As SqlDataAdapter
    da = New SqlDataAdapter(cmd)
    da.SelectCommand = cmd

    Dim ds As DataSet
    ds = New DataSet()
    da.Fill(ds, "Schema")
    Return ds
End Function
```

Listing 1 A built-in table called INFORMATION_SCHEMA contains information about the schema of all the tables that belong to the database the application is connected to.

GetOleDbSchemaTable:

```
Public Function GetOleDbSchemaTable(_
    ByVal schema As Guid, _
    ByVal restrict() As Object) As DataTable
```

The method takes two arguments and returns a DataTable object populated with all the requested schema information. The first argument serves to identify which category of schema information you're asking for. The schema category is an element of type OleDbSchemaGuid. The OleDbSchemaGuid class contains only static members, each of which maps to a predefined OLE DB schema rowset including catalogs, tables, columns, constraints by table, and more. For example, by setting the schema argument to OleDbSchemaGuid.Catalogs, you instruct the OLE DB provider to return a list of all the catalogs in the specified instance of the DBMS:

```
dt = conn.GetOleDbSchemaTable(_
    OleDbSchemaGuid.Catalogs, Nothing)
```

The connection string plays a key role because the GetOleDbSchemaTable works on top of an open connection. First, make sure the connection string includes the PROVIDER token so you can locate the desired OLE DB provider. Use this call to get all the SQL Server's catalogs on the local machine:

```
s = "SERVER=...;UID=...;"
s += "PROVIDER=sqloledb;"
```

```
conn = New OleDbConnection(s)
conn.Open()
```

Next, if you want to query a particular catalog, you can set its name in the connection string through the DATABASE token. However, the connection string doesn't allow you to restrict the amount of information returned in a flexible way.

Using the second argument of the GetOleDbSchemaTable method results in much more powerful code. The *restrict* argument is an array of Object types whose size depends on the schema GUID you selected with the first argument. The argument acts as a kind of WHERE clause for all the elements in the specified schema type's restriction table. A restriction table comprises the parameters you can filter the schema information on. Catalogs, Tables, Columns, and all the possible other members of the OleDbSchemaGuid class have their restriction tables. Use this code to get column information for the MyData.MyClients table:

```
dt = conn.GetOleDbSchemaTable(_
    OleDbSchemaGuid.Catalogs, filters)
```

The filters argument is an array of Object types with four elements, because the Columns restriction table contains four elements: catalog, schema, table, and column name:

```
Dim filters(3) As Object
filters(0) = "MyData"
filters(1) = Nothing
filters(2) = "MyClients"
filters(3) = Nothing
```

This code sets the command to query for schema information about all the columns in the MyClients table in the MyData catalog. You could restrict the resultset to only the specified column by setting the fourth element to a column name. Pay attention to the array's size; it must always be no greater than the size of the element's restriction table. (See the .NET Framework documentation to learn about the characteristics of each restriction table.) The GetOleDbSchemaTable method returns a DataTable whose number of columns cannot be filtered but includes as much information as you need.

Not all OLE DB providers might support all the schema types the method can take in. In this case, you're returned an empty DataTable.

Understand ADOX

Both SQL and OLE DB share a common approach to getting schema information. Both execute low-level commands and, to some extent, force you to reason in a low-level way. You can obtain any piece of schema information only through the underlying database, but this doesn't mean you must work in this way. OLE DB provides a first level of abstraction and allows you to call only one method, irrespective of the physical data source's intricacies. The rub with OLE DB lies in the fact that it treats schema information as read-only, and you can't alter the database's state. Although better than nothing, this isn't exactly what people would hope for. You might think you could resort to ADOX if you want to use an object model to issue DDL statements.

ADOX is an extension to the ADO objects that includes objects for schema creation and modification, as well as security. ADOX lets you write object-based code that works against various data sources, regardless of differences in their native syntaxes. With ADOX, creating tables and columns is as easy as creating new instances of existing objects. Among other things, ADOX makes fulfilling any business logic seamless while initializing or modifying the data layer's structure.

Unfortunately, ADOX has several limitations when called to work with ODBC and OLE DB providers other than the Jet OLE DB Provider. This is because ADOX itself is designed primarily for use with Access databases. However, although ADOX has some limitations (that is, Jet's notion of a catalog doesn't match SQL Server's), you can also read and write schema information on a SQL Server table.

Using ADOX from within a .NET application is harder than using ADO. You import the ADOX library into the project and start using the objects you need. One caveat: You can't use ADO.NET connection objects. An ADOX catalog recognizes only ADO's connection objects. If you try to bind the ADOX library to an OleDbConnection object, an exception is thrown because ADOX queries for a missing interface. You must import ADO in order to use ADOX.

ADO.NET has no special support yet for schema manipulation. As of today, you must endeavor to find the best workaround that fixes your particular problem. The most powerful is the first I've shown here—that is, using the provider-specific DDL instructions. If you don't need to do complex tasks, consider ADOX, but also consider that ADOX affects performance due to its COM nature. However, this might not be a big issue for a one-shot call. OLE DB provides a .NET-specific, higher-level approach, but doesn't allow you to make schema changes to the database.

According to announcements made and tech-preview demos shown at Microsoft's Professional Developers Conference (PDC) 2001, Microsoft is committed to improving the ADO.NET model in the next major release of the .NET Framework. Although details are subject to change, Microsoft plans to address the lack of powerful and native tools for schema manipulation with a new ADOX-like object model. This would allow you to create catalogs and tables using an object-based approach with no limitations. The idea is to add a new group of classes to the specification of .NET data providers so each one can provide object-oriented replacements for today's DDL statements. **VSM**

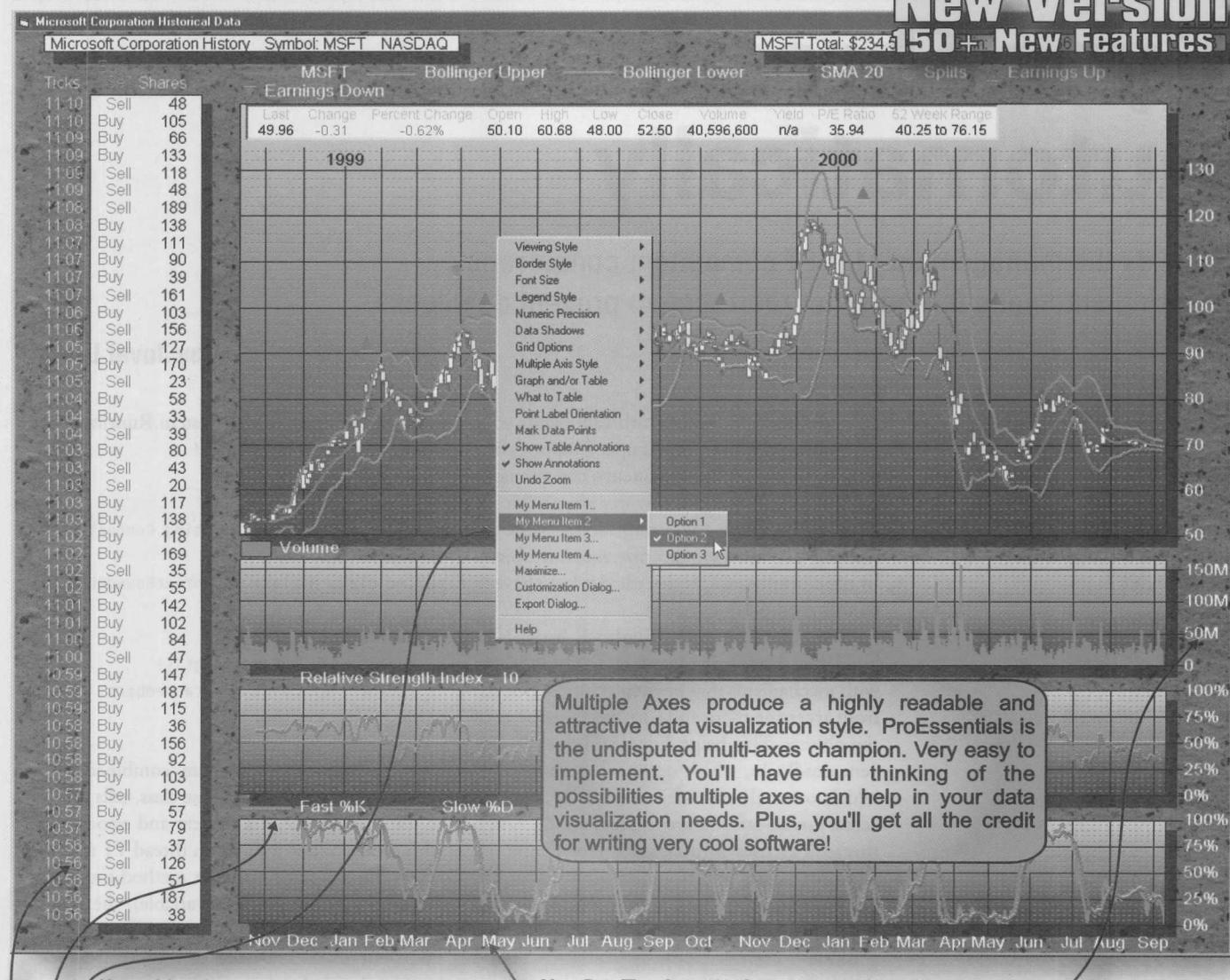
Dino Esposito is a trainer and consultant for Wintellect (www.wintellect.com). He writes for *MSDN Magazine* and *MSDN Voices*. The author of *Building Web Solutions with ASP.NET and ADO.NET* (Microsoft Press) and of the upcoming *Applied XML Programming for Microsoft .NET*, Dino is also the cofounder of www.vb2themax.com. Reach him at dinoe@wintellect.com.

Additional Resources

Microsoft ADO Extensions 2.1 for DDL and Security Object Model: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/modcore/html/deovrmicrosoftadoextensions21forddlsecurity.asp>

GigaSoft® ProEssentials™ v4.0

New Version
150+ New Features



New quick and easy mechanism to control and extend the built-in popup-menu system.

More Date/Time formatting features to complement the most elaborate date/time capabilities of any charting tool in the market.

New Legend Styles and per axis Legend Locations. New Separated Multiple Axes with per axis Gradients, justifiable BitBlts, and AlphaBlend features. Plus three new Border Types to apply to axes and tables.

New Tick Styles and Tick Color Control. New event driven grid number formatting. New optional scaling logic for grid number labels.

New extensive multiple table annotation mechanism allowing the robust mixture of text and symbols within the chart with options for real-time update, hot spots, header columns, and header rows.

Undisputed #1
Best Multiple Axes
Best Quantity of Features
Best Client Side Functionality

The real power of .NET is WebServices, and ProEssentials is the perfect match. Combining the best client side charting with .NET's best server feature gives you the best of both worlds. The future is evolving towards powerful client side internet enabled software!

For over 7 years, programmer's worldwide have recognized GigaSoft Inc. as a leading innovator in charting functionality. Here's just a few of the charting features which ProEssentials brought to market first:

- 1st Charting Tool to table data below the chart.
- 1st Charting Tool to support multiple y axes.
- 1st Charting Tool to include dynamic date/time handling.
- 1st Charting Tool to support overlapping multiple axes.
- 1st Charting Tool to support per axis gradients, bitblts, and alphablends.
- 1st Charting Tool to include table annotations and 27 viewing styles.
- Plus many more firsts, all stably implemented and easy to use.

OCX : DLL : VCL soon to come... WinForms : WebForms

ProEssentials v4 including subscription and technical support \$999.00 Upgrade from v3 \$499.00 Subscription includes native .NET interfaces due middle 2002. Includes 1 CPU/Server license for Web Server Applications, Royalty Free for Distributed Applications.

gigasoft.com

GigaSoft, Inc.
Phone: 817 431-8470
Fax: 817 431-9860
Email: info@gigasoft.com

Copyright 2002 GigaSoft, Inc. GigaSoft is a registered trademark and ProEssentials a trademark of GigaSoft, Inc. All other trademarks or registered trademarks are the respective property of their owners.

Sync Threads Automatically

Simplify the management of .NET component concurrency by using automatic thread synchronization to prevent deadlocks.

by Juval Löwy

Technology Toolbox

- VB.NET**
- C#**
- SQL Server 2000**
- ASP.NET**
- XML**
- VB6**
- Other:**
.NET Framework

Go Online!

Use these Locator+ codes at www.visualstudiomagazine.com to go directly to these related resources.

Download

VS0209BB_T Download the code for this article, which includes a sample component that uses a synchronization domain and a test client, both provided in C# and VB.NET.

Discuss

VS0209BB_D Discuss this article in the .NET forum.

Read More

VS0209BB_T Read this article online. It includes inline code in both C# and VB.NET.

VS0202BB_T Black Belt, "Demystify .NET App Domains and Contexts," by Juval Löwy

N020213CS "Make Synchronization Automatic" by Juval Löwy

The introduction of multithreading into your VB6 application opens up a Pandora's box of synchronization and concurrency management issues. You have to worry about threads deadlocking while contesting for the same resources. You must synchronize access to objects by concurrent multiple threads. And you have to handle method re-entrance.

Your first step in tackling such issues is to build thread-safe components by equipping them with mechanisms that prevent multiple threads from accessing them and corrupting the state of the objects. This helps, but it still doesn't prevent deadlocks, which occur when thread T1, which owns thread-safe resource R1, tries to access thread-safe resource R2 just as R2's owner, thread T2, tries to access R1 (see Figure 1). Multithreading defects are notoriously hard to isolate, reproduce, and eliminate. They often involve rare race conditions, and fixing one problem often introduces another. Before .NET, it was nontrivial to write robust, high-performance, multithreaded code. You needed a great deal of skill and discipline to succeed.

Enter .NET, which aims at simplifying component concurrency management. By default, all .NET components execute in a multithreaded environment that allows concurrent access by multiple threads. I'll show you how to use .NET's automatic synchronization, which lets you decorate your component with an attribute and have .NET manage concurrent access to the object.

Automatic synchronization revolves around intercepting calls coming into a component's context (see the Read More section in the Go Online box for more information on app domains and contexts). Components must be context-bound to take advantage of .NET automatic synchronization. This means you need to constrain them to execute always in the same context. These components must derive from the ContextBoundObject class directly or indirectly, and they must use the Synchronization

attribute, defined in the System.Runtime.Remoting.Contexts namespace:

```
//C#
using System.Runtime.Remoting.Contexts;
[Synchronization]
public class MyClass : ContextBoundObject
{
    public MyClass(){}
    public void DoSomething(){}
    //other methods and data members
}
```

The Synchronization attribute, combined with the ContextBoundObject base class, tells .NET to place the object in a context and associate it with a lock. When a client on thread T1 tries to access the object by calling a method on it (or accessing a public member variable), the client actually interacts with a proxy. .NET intercepts the client access and tries to acquire the lock associated with the object. If the lock isn't owned by another thread currently, .NET acquires the lock and accesses the object on thread T1. .NET

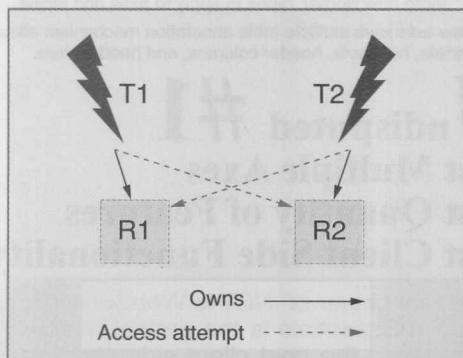


Figure 1 Multithreading Can Be Deadly. A deadlock can occur when two or more threads each own a resource and is waiting for a resource owned by another thread. Deadlocks are notoriously hard to resolve and often appear unpredictably.

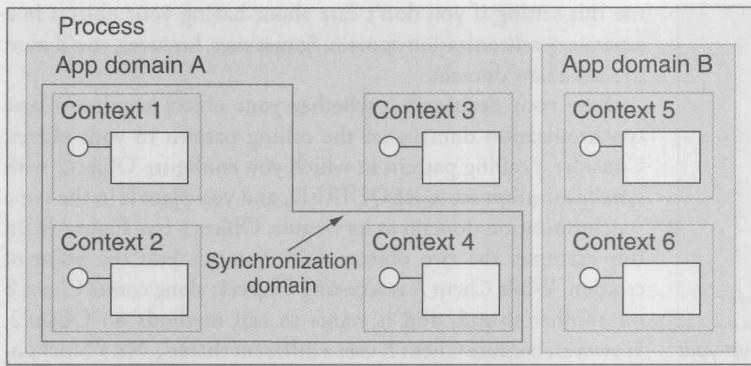


Figure 2 Synchronize Domains, Contexts, and App Domains. The .NET Framework's synchronization domains can save you from error-prone manual thread synchronization. A synchronization domain is independent of context, but is limited to a single app domain. A context belongs to one synchronization domain at most.

releases the lock and returns control to the client when the call returns from the method. However, if another thread T2 were accessing the object, then T1 would be blocked until T2 releases the lock. In fact, all other threads are placed in a queue while the object is accessed by one thread, so they get to access the object in order one at a time.

Avoid Deadlocks With Shared Locks

.NET could have allocated one lock per context-bound object, but that would have been inefficient. Objects can often share a lock and execute on the same thread—if by design the components are all meant to participate in the same activity on behalf of a client. In such situations, allocating one lock per object would waste resources and processing time, forcing .NET to do additional locks and unlocks on every object access. Moreover, sharing locks among objects would reduce the likelihood of deadlocks. If two objects interact with each other and each has its own lock, two different clients on different threads can use these objects. The objects would then deadlock when they try to access each other. If the objects were to share a lock, only one client thread would be allowed to access them.

In .NET, a set of context-bound objects sharing a lock belongs to a *synchronization domain*. Each domain has one lock; multiple threads can't make concurrent calls within the same synchronization domain. When a thread accesses one object in a synchronization domain, that thread can access the other objects in the domain. In fact, the

synchronization domain locks all its objects from access by other threads, even though the current thread in the domain accesses only one object at a time.

Synchronization domains are context-independent and can include objects from multiple contexts. However, a synchronization domain is limited to a single app domain: Objects from different app domains can't share a synchronization domain lock. A context can belong to no more than one synchronization domain at a time, if any. If a context belongs to a synchronization domain, then all the objects in that context belong to that synchronization domain (see Figure 2).

You need to decide how to associate a component with a synchronization domain lock. You can choose whether the object needs a lock at all, whether it can share a lock with other objects, or whether it requires a new lock. The *SynchronizationAttribute* class provides a number of overloaded constructors, all accepting a constant integer flag. Possible values for the flag are NOT_SUPPORTED, SUPPORTED, REQUIRED, and REQUIRES_NEW. You use these constants to *allocate* an object to a synchronization domain relative to its creating client:

```
//C#
[Synchronization(SynchronizationAttribute.REQUIRES_NEW)]
public class MyClass : ContextBoundObject
{}
```

The default constructor of the *SynchronizationAttribute* class uses REQUIRED. .NET gives you three options. First, you can place an object in its creator's synchronization domain, in which case the object shares a lock with its creator. Second, you can place an object in a new synchronization domain, where the object has its own lock and starts a new synchronization domain. Finally, you can choose to not place the object in a synchronization domain, in which case you get concurrent access and no lock.

Pick a Sync Domain

.NET determines an object's synchronization domain at creation time, based on the synchronization domain of its creator and the constant value you choose for the *Synchronization* attribute (see Table 1). That's because .NET uses a heuristic, which assumes

| Synchronization constant flag | Does the object's creator have a synchronization domain? | Synchronization domain determined by .NET |
|-------------------------------|--|--|
| NOT SUPPORTED | — | The object will never be part of a synchronization domain, regardless of whether its creator has a synchronization domain. |
| SUPPORTED | Yes | .NET places the object in its creator's synchronization domain. |
| SUPPORTED | No | The newly created object doesn't have a synchronization domain. |
| REQUIRED | Yes | .NET puts the object in its creator's synchronization domain. |
| REQUIRED | No | .NET creates a new synchronization domain for the object. |
| REQUIRES NEW | — | .NET creates a new synchronization domain for the object, regardless of whether its creator has a synchronization domain. |

Table 1 Determine Your Object's Synchronization Domain. .NET determines an object's synchronization domain at creation time, based on the synchronization domain of its creator and the constant value you provide for the *Synchronization* attribute. REQUIRED is the default for the *Synchronization* Attribute class.

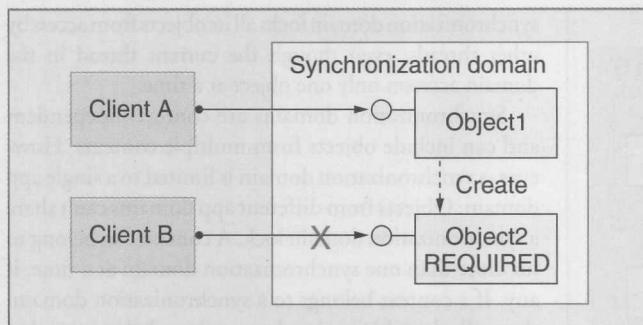


Figure 3 Share a Lock With the Creator. Determine whether your object requires a new synchronization domain based on the calling pattern to your object. Consider this calling pattern, in which you configure Object2 with synchronization set to REQUIRED, and you place it in the same synchronization domain as its creator, Object1. Sharing the lock with the creator when the two objects don't interact causes Client B to wait until the first call is completed, even though it could access the object safely.

that calling patterns, interactions, and synchronization needs between objects usually resemble the relationship between an object and its creator.

The various Synchronization attribute construction values give you a variety of options. An object with synchronization set to NOT_SUPPORTED never participates in a synchronization domain. The object must provide its own synchronization mechanism. Use this setting only if you expect concurrent access, and you want to provide your own synchronization mechanisms. But why do that? Context-bound objects should leverage .NET's component services support.

An object with synchronization set to SUPPORTED shares its creator's synchronization domain if it has one, and has no synchronization of its own if the creator doesn't have one. Use SUPPORTED for the rare case when the component itself has no need for synchronization, but downstream objects it creates do require it. Components with synchronization SUPPORTED can propagate the synchronization domain of their creating client to downstream objects, which then share one synchronization domain instead of having separate ones. This reduces the likelihood of deadlocks.

You most often set object synchronization to REQUIRED—that's why it's the SynchronizationAttribute class default. Always

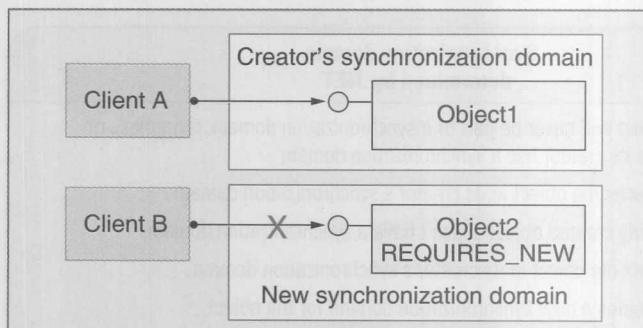


Figure 4 Start a New Synchronization Domain. In this calling pattern, having a synchronization domain separate from the created object enables the object to service its clients more efficiently. Using class factories to create objects provides a classic example of needing to configure components to require a new synchronization domain.

use this setting if you don't care about having your objects in a separate synchronization domain. Sometimes, however, you'll want to start a new domain.

Base your decision on whether your object requires a new synchronization domain on the calling pattern to your object. Consider a calling pattern in which you configure Object2 with synchronization set to REQUIRED, and you place it in the same synchronization domain as its creator, Object1 (see Figure 3). In this example, the two objects don't interact past the point of creation. While Client A is accessing Object1, along comes Client B on another thread, and it wants to call methods on Object2. However, because Client B uses a different thread, .NET blocks it, even though it could have accessed Object2 safely, because it doesn't violate the synchronization requirement for the creating object, Object1.

Class Factories Need New Domains

On the other hand, if you were to configure Object2 to require its own synchronization domain by using REQUIRES_NEW, the object could process calls from other clients at the same time as Object1 (see Figure 4). Using class factories to create objects provides a classic example of needing to configure components to require a new synchronization domain.

Class factories usually require thread safety because they service multiple clients. Once a factory creates an object, though, it hands the object back to a client and has nothing more to do with it. You need to configure the objects to require a new synchronization domain because you don't want all the objects created by a class factory to share the same synchronization domain.

However, calls from the creator object (Object1) to Object2 will now potentially block and will be more expensive because the calls must cross context boundaries and pay the overhead of trying to acquire the lock. You can synchronize context-bound objects most easily using .NET synchronization domains. These provide a modern synchronization technique that formally eliminates many synchronization problems and the consequent need to code around them, then test the handcrafted solution.

Synchronization domains provide a substantial productivity gain, but you do need to consider four limitations. First, you can use synchronization domains only with context-bound objects. All other .NET types require manual synchronization objects. Second, you could have performance issues when you access context-bound objects using proxies and interceptors in an intense calling pattern. Third, synchronization domains don't protect static class members and static methods. Those require manual synchronization objects. Finally, synchronization domains are not throughput-oriented. An incoming thread locks a set of objects even if it interacts with only one. This precludes other threads from accessing these objects, and theoretically could degrade application throughput.

For balance, you must rely on synchronization domains and other advanced component services in any decent-sized application—or whenever productivity and quality are top priorities. **VSM**

Juval Löwy is a software architect and principal of IDEsign, a consulting and training company focused on .NET. This article derives from his upcoming book on programming .NET components (O'Reilly). Juval speaks at development conferences and chairs the .NET California Bay Area User Group's program committee. Contact him at www.idesign.net.

Every article ever published in *Visual Studio Magazine* is available online for free. The editors of *VSM* compiled this index to assist you in finding articles that answer your burning questions about how Visual Studio can solve your business problems. Use the keywords to identify the articles that address the topics you're interested in, then go to www.visualstudiomagazine.com and enter the associated Locator+ code in the field in the upper right-hand corner of the page.

| Title | Author | Keywords | Locator+ Code |
|--|---|--|---------------|
| September 2002 | | | |
| Debug VS.NET Applications | Nancy Folsom and Kathleen Dollard | VB.NET, C#, debug | VS0209NF_T |
| Getting Started: Manage Contacts With Outlook and VB | Stan Schultes | VB6, Outlook 2002, Outlook 2002, ADO | VS0209GS_T |
| Desktop Developer: Deploy WinForms on the Web | Mauro Sant'Anna | VB.NET, WinForms | VS0209DT_T |
| Q&A: Keep a Window Visible | Karl E. Peterson, Juval Löwy, and Mattias Sjögren | VB.NET, C#, VB6, VB5, moving windows offscreen, unsafe code, constant object variables | VS0209QA_T |
| ASP.NET: Create Boilerplate Templates | Jonathan Goodyear | C#, ASP.NET, VS.NET, customize code, boilerplate code | VS0209AN_T |
| Database Design: Manipulate Schema Information | Dino Esposito | VB.NET, SQL Server 2000, schema manipulation, ADOX, OLE DB, ADO.NET | VS0209DD_T |
| Black Belt: Sync Threads Automatically | Juval Löwy | VB.NET, C#, .NET Framework, automatic thread synchronization | VS0209BB_T |
| Create Localizable Web User Controls | Gerardo Villeda | VB.NET, Web control, cultural settings | VS0209GV_T |
| Oracle Provider for .NET Extends ADO.NET | Lee Thé | George Demarest, Oracle, Windows, .NET | VS0209LT_T |
| Increase Availability and Performance With Threads | Edward G. Nilges | VB.NET, threading | VS0209EG_T |
| Editor's Note: Goin' Mobile With MMIT | Patrick Meader | Mobile Internet Toolkit, MIT, Compact Framework, Palm, ASP, ASP.NET | VS0209EN_T |
| Guest Opinion: Pick Leaders Who Can Help You | Nancy Folsom | VB.NET, .NET, debate, leaders | VS0209GO_T |
| August 2002 | | | |
| Speed Up Your VB.NET Code | Francesco Balena | VB.NET, optimization, eight tips | VS0208FB_T |
| Design Custom Charts | Jonny Anderson | C#, GDI+, accelerate business charts | VS0208JA_T |
| Upgrade to Firewall-Friendly Updategrams | Roger Jennings | SQL Server 2000, XML, VB6, MSDE, IIS, Windows 2000, updategrams, secure transactions | VS0208RJ_T |
| Getting Started: Exploit the FileSystemObject Class | Andy Rosebrock | VB.NET, C#, VB6, file management object model, file-system tasks | VS0208GS_T |
| Desktop Developer: Draw Asynchronously With .NET | Bill Storage | VB.NET, C#, GDI+, background thread, multithreading | VS0208DT_T |
| Q&A: Handle API Callbacks Safely | Karl E. Peterson and Juval Löwy | VB.NET, C#, VB6, VB5, API callbacks, assemblies, compilation warnings | VS0208QA_T |
| ASP.NET: Add Google Search to Your Site | Boris Feldman | VB.NET, ASP.NET, XML, .NET Framework, GoogleBox, user control, Google search | VS0208AN_T |
| Database Design: Save Database Trips | Bill Wagner | C#, SQL Server 2000, ADO.NET, System.Data.DataSet class, database code creation | VS0208DD_T |
| Black Belt: Look Sharp at Design Time | Bill Storage | C#, custom designers | VS0208BB_T |
| Secure Your Web Apps | Timothy M. Chester | ASP.NET, forms authentication, XML | VS0208TC_T |
| Take Charge of Garbage Collection | Mauro Sant'Anna | garbage collection, freeing resources | VS0208MS_T |
| Leverage Current Apps With Web Services | Patrick Meader | Infragistics, Dean Guida, Web services, .NET, C#, VB.NET, copy protection, Java | VS0208PM_T |
| Editor's Note: VB6 Today, Tomorrow, Next Week | Patrick Meader | VB6, VB.NET, Microsoft support, mainstream phase, extended phase | VS0208EN_T |
| Guest Opinion: Office and .NET: Better Together? | Jonathan West | Office, VBA, VSA, .NET strategy | VS0208GO_T |
| July 2002 | | | |
| Build a Scalable Architecture | Kathleen Dollard | VB.NET, VB6, ADO.NET, n-tier | VS0207KD_T |
| Create a Trickle-Feed Application | Zane Thomas | C#, .NET assemblies, application domains | VS0207ZT_T |
| Getting Started: Create and Consume XML Web Services | Stan Schultes | VB.NET, ASP.NET, XML, VB6, Web services, WinForms, SOAP, Web Forms | VS0207GS_T |
| Desktop Developer: Evaluate Strings and Convert Data in .NET | Edward G. Nilges | VB.NET, strings, data conversion | VS0207DT_T |

| Title | Author | Keywords | Locator+ Code |
|--|--------------------------------------|---|----------------------|
| Q&A: Drag and Drop URLs | Karl E. Peterson and Juval Löwy | C#, VB6, VB5, custom collections, OLE drag-and-drop, URLs | VS0207QA_T |
| ASP.NET: Implement Client Scripts | Ken Cox | VB.NET, ASP.NET, client-side scripts | VS0207AN_T |
| Database Design: Compare XML Data Access APIs | Greg Ewing and Jon Rauschenberger | C#, SQL Server 2000, ASP.NET, VB6, XML, data access | VS0207DD_T |
| Black Belt: Exploit .NET Objects | Alberto Falossi | VB.NET, C#, .NET objects | VS0207BB_T |
| Create a Data Upload Service | Matt Hostetler | ASP, ADO, XML, XSLT, data upload, SQL Server 2000, VB6, SOAP Toolkit | VS0207MH_T |
| Web Services Interop to Improve Security | Lee Thé | Web Services Interoperability Organization, security, Neil Charney, Chris Kurz | VS0207LT_T |
| Detect Windows Version Directly | Hank Marquis | VB6, VB.NET, Windows OS, Windows version detection | VS0207HM_T |
| Editor's Note: Visit Our New Home | Patrick Meader | Fawcette Technical Publications, online catalog, www.visualstudiomagazine.com | VS0207EN_T |
| Guest Opinion: .NET Migration Works—Kind Of | Bill Wagner | .NET migration, .NET Framework, Managed C++ | VS0207GO_T |
| June 2002 | | | |
| Subclass Controls in .NET | Bill Storage | VB.NET, subclass controls, custom controls, owner-draw listbox | VS0206BS_T |
| Salaries Stabilize in Tough Market | Susannah Pfalzer | salary survey, salaries, economy | VS0206SP_T |
| Getting Started: Create Random Access Log Files | Billy Hollis | VB6, log files, random access, user-defined types | VS0206GS_T |
| Desktop Developer: Craft .NET Controls With Inheritance | Michael Welch | VB.NET, C#, .NET controls, inheritance, custom controls | VS0206DT_T |
| Q&A: Develop Rich-UI Apps | Karl E. Peterson and Juval Löwy | C#, VB6, Windows Forms, MDI, MFC, AssemblyVersion, Microsoft Word | VS0206QA_T |
| ASP.NET: Provide Rich Functionality With Server Controls | Chris Kinsman | C#, ASP.NET, server controls | VS0206AN_T |
| Database Design: Create More Efficient Database Code | Dino Esposito | SQL Server 2000, ADO.NET, .NET Data Provider, database code | VS0206DD_T |
| Black Belt: Decouple Components With LCE | Juval Löwy | C#, loosely coupled events, LCE, decouple | VS0206BB_T |
| Exploit Binary Compatibility With .NET Metadata | Juval Löwy | C#, C++, metadata, binary compatibility | VS0206JL_T |
| Build Plug-and-Play Web Portals | Lee Thé | Web Services for Remote Portals, WSRP, plug and play, OASIS, Thomas Schaeck | VS0206LT_T |
| Improve ASP.NET Application Performance | Paul Delcogliano | VB.NET, SQL Server 2000, ASP.NET, XML, IIS, IE5, Java Script, Web services | VS0206PD_T |
| Secure Your Web Services | Wei Meng Lee | VB.NET, ASP.NET, XML, SOAP, .NET Framework, Web services, encryption | VS0206WL_T |
| Editor's Note: Real-World .NET | Patrick Meader | Visual Studio .NET, VB.NET, C#, ASP.NET, corporate applications | VS0206EN_T |
| Guest Opinion: VB.NET = .NOT ENOUGH | Bill Storage | VB.NET, productivity | VS0206GO_T |
| May 2002 | | | |
| Invoke Asynchronous Magic | Robert Teixeira | VB.NET, background tasks, asynchronous mode | VS0205RT_T |
| Access Windows Networking Info | L.J. Johnson | VB6, networking, Windows 9x | VS0205LJ_T |
| Enhance SQL Server Security | Walter Myers III and David Byres | SQL Server 2000, VB6, ASP, ADO, IIS, Windows 2000, VBScript, security | VS0205MB_T |
| Getting Started: Master .NET's New String Objects | Stan Schultes | VB.NET, string objects, .NET Framework, StringBuilder | VS0205GS_T |
| Desktop Developer: Detect Your Windows Version | Hank Marquis | VB.NET, VB6, Windows version | VS0205DT_T |
| Q&A: Retrieve Command-Line Arguments | Karl E. Peterson and Mattias Sjögren | C#, VB6, command line, GetUserName, CD AutoRun, System.Environment | VS0205QA_T |
| ASP.NET: Expand and Collapse Web DataGrids | Dino Esposito | C#, ASP.NET, DataGrid | VS0205AN_T |
| Database Design: Display Enterprise Reports Remotely | Josh Lane | C#, SQL Server 2000, IIS, Excel 2000, Northwind, Microsoft Foundation Classes, enterprise reports | VS0205DD_T |
| Black Belt: Get Debug Info From Production Apps | Dan Fergus | VB6, PDB file, debugging | VS0205BB_T |
| Editor's Note: Trusting Security to Microsoft | Patrick Meader | security, IIS | VS0205EN_T |
| Guest Opinion: Web Services Dodge CORBA's Mistake | Jon Rauschenberger | COBRA, Web services, WSDL | VS0205GO_T |

| Title | Author | Keywords | Locator+ Code |
|--|--------------------------------------|--|----------------------|
| Buyers Guide 2002 | | | |
| .NET Compact Framework | Dan Fergus | .NET Compact Framework, Editor's Choice | VS02BGDF_T |
| New .NET Tools Lower the Learning Curve | Don Kiely | VB.NET, C#, .NET tools, third-party components | VS02BGDK_T |
| How to Choose Web Services | Jonathan Goodyear | Web services, Web service vendors, SOAP | VS02BGJG_T |
| Software Vendors Boost .NET | Don Kiely | software vendors, VS Live!, .NET | N020226VS |
| Vendors Debut .NET Products | Susannah Pfalzer | software products, VS Live! | N020214EX |
| RAD Add-Ins Lead the .NET Pack at VS Live! | Roger Jennings | RAD, VS Live! | N020213RJ |
| Extending Visual Studio .NET | Patrick Meader | .NET panel discussion, VS Live! | N020214RT |
| Editor's Note: VS.NET: Integrate Seamlessly | Patrick Meader | Visual Studio .NET, seamless integration, extensibility | VS02BGEN_T |
| Guest Opinion: Speed Kills | Josh Lane | performance, code speed | VS02BGGO_T |
| April 2002 | | | |
| Call VB6 From .NET | Rockford Lhotka | VB.NET, VB6, Interop service, .NET DLLs | VS0204RL_T |
| Power Up Database Apps With XML | Daniel Anderson and Mary V. Hooke | SQL Server 2000, XML, VB6, database apps, IIS, MSXML parser | VS0204DA_T |
| Tame .NET Events | Juval Löwy | C#, .NET events, event-handling | VS0204JL_T |
| Getting Started: Build a Custom Grid Collection | Stan Schultes | VB6, custom classes, collection class, grid collection | VS0204GS_T |
| Desktop Developer: Link Data to Your UI Automatically | Bill Wagner | VB.NET, C#, data linking, data model, .NET Framework, data binding | VS0204DT_T |
| Q&A: Restore Tray Icons After Crashes | Karl E. Peterson and Brian Noyes | C#, XML, VB6, Windows Explorer, graphics, XML fragments | VS0204QA_T |
| ASP.NET: Serve Dynamic Pages Quickly | Bill Wagner | VB.NET, SQL Server 2000, ASP.NET, DataSet class, dynamic pages | VS0204AN_T |
| Database Design: Whip Your Queries Into Shape | Andy Clark | VB.NET, SQL Server 2000, VB6, MSDataShape Provider, ADO, data queries | VS0204DD_T |
| Black Belt: Trace and Cure Code Ailments | Lars Powers and Mike Snell | VB.NET, C#, debugging, tracing | VS0204BB_T |
| A Talk With Database Guru Jim Gray | Lee Thé | Jim Gray, SQL Server 2000 | N020205JG |
| Scale .NET Apps With COM+ | Keith Franklin | VB.NET, C#, SQL Server 7, scalability, COM+ | VS0204KF_T |
| Get More Accurate Data With NULLs | Josef Finsel | SQL Server, NULL, database values | VS0204JF_T |
| 6 Great Tips for Designing Data Components with .NET in Mind | Roger Jennings | data components, .NET Framework | N011127SS |
| Editor's Note: Clearing Up .NET Misunderstandings | Patrick Meader | Visual Studio .NET, .NET Framework, VB6, COM, API, OOP, Web services | VS0204EN_T |
| Guest Opinion: IDE Gives Sight to .NET Vision | Dan Fergus | IDE, Web services, Web forms, VS.NET | VS0204GO_T |
| March 2002 | | | |
| Smash SQL Speed Barriers | Dan Fox | VB.NET, SQL Server 2000, ADO.NET, SqlCommand, ODBC, OLE DB provider, .NET Framework | VS0203DF_T |
| Getting Started: Extend Your About Box | Stan Schultes | VB.NET, System.Reflection, About boxes | VS0203GS_T |
| Desktop Developer: Implement Versioning in .NET | Bill Wagner | VB.NET, C#, .NET Framework Library, versioning | VS0203DT_T |
| Q&A: Name That Message | Karl E. Peterson and Robert Teixeira | VB.NET, VB6, message name, .NET Framework, ping functionality | VS0203QA_T |
| ASP.NET: Create Custom Performance Counters | Jonathan Goodyear | C#, ASP.NET, .NET Framework, IIS, Visual Studio .NET, performance counters | VS0203AN_T |
| Web Services: Build Scalable Web Services With MSMQ | Josh Lane | VB.NET, SQL Server 2000, ASP.NET, Windows XP Professional, Web services, MSMQ, scalable services | VS0203WS_T |
| Database Design: Sync VB Code With a Database | Michael Domingo | SQL Server, VB6, ADO, SQL-DMO, code synchronization, VB Extensibility | VS0203DD_T |
| Black Belt: Safeguard Data in VB6 Collections | Dan Fergus | VB6, collections, type safety | VS0203BB_T |
| Add Object Models to Non-COM Apps | Leonardo Esposito | C++, VBScript, object models, non-COM apps | VS0203LE_T |
| Keep Your Projects From Failing | Eds. | Jan Popkin, Popkin Software, project failures, modeling | N020108VS |
| Back to Basics: XML Parsing APIs | Dan Wahlin | XML parsing, APIs | N020108XM |
| Run Full-Text Database Searches On a Shoestring | Roger Jennings | database searches | N011113SS |
| Connect Office Apps to XML Web Services | Roger Jennings | XML, Web services, Microsoft Office | N020102SS |

| Title | Author | Keywords | Locator+ Code |
|---|-----------------------------------|---|---------------|
| Editor's Note: What's in It for You? | Patrick Meader | Visual Studio .NET, ADO.NET, IDE | VS0203EN_T |
| Guest Opinion: Hire Essential .NET Talent | Josh Lane | hiring, software engineering, accidental skills, essential skills | VS0203GO_T |
| February 2002 | | | |
| Describe Code With Class | Robert Teixeira | VB.NET, C#, .NET beta 2, code description | VS0202RT_T |
| Getting Started: Take Data Presentation to the Next Level | Andy Rosebrock | VB6, DataRepeater control, data presentation, UI | VS0202GS_T |
| Desktop Developer: Enhance the Printers Collection | Karl E. Peterson | VB6, VB5, API calls, Printers collection | VS0202DT_T |
| Q&A: Provide Custom Hierarchical URLs | Jonathan Goodyear and Robert Lair | VB.NET, C#, ASP.NET, XML, URL structure, AdRotator control | VS0202QA_T |
| ASP.NET: Debug Your ASP.NET Apps Using Trace | Scott Jamison | VB.NET, ASP.NET, Visual Studio .NET, debugging, TraceContext class | VS0202AN_T |
| Web Services: Increase Web Service Performance | Tim Chester | C#, ASP.NET, XML, Access 2000, Web service performance | VS0202WS_T |
| Database Design: Search Databases the .NET Way | Andy Clark | VB.NET, SQL Server 2000, database searching | VS0202DD_T |
| Black Belt: Demystify .NET App Domains and Contexts | Juval Löwy | VB.NET, C#, contexts, app domains, common language runtime, JIT | VS0202BB_T |
| Pick the Right Tool for the Job | Bill Wagner | .NET tools | VS0202BW_T |
| Perform Multiple Tasks Using Threads | Bill Wagner | C#, threads, multitasking | VS0202BL_T |
| Is VS.NET Ready for the Enterprise? | Lee Thé | Rational Software, Mike Devlin, enterprise solutions, Visual Studio .NET | VS0202LT_T |
| Face Off: Extreme Programming vs. Interaction Design | Elden Nelson | extreme programming, interaction design, Kent Beck, Alan Cooper | N020115BC |
| Customize the IIS Error Page | Jonathan Goodyear | ASP, Visual InterDev 6, IIS errors | VS0202JG_T |
| Editor's Note: Judging Web Services | Patrick Meader | Web services, contest | VS0202EN_T |
| Guest Opinion: Catch a Rising Star | Henry G. Fiorentini III | .NET technology, transition | VS0202GO_T |
| January 2002 | | | |
| Retrieve Data Easily | Andrew J. Brust | VB.NET, SQL Server 2000, ASP.NET, XML, ADO.NET, Web Forms, Windows Forms, data retrieval | VS0201AB_T |
| Add File Replication to Your App | Monte Hansen | VB6, IE4, OO interface, Win32 API, file access | VS0201MH_T |
| Build an XML Serialized Object | Jonathan Goodyear | VB.NET, ASP.NET, XML, .NET Framework SDK, IIS, XML serialized object | VS0201JG_T |
| Getting Started: Understand Visual Inheritance | Stan Schultes | VB.NET, visual inheritance, WinForms | VS0201GS_T |
| Desktop Developer: Communicate With Windows Services | Billy Hollis | VB.NET, Windows 2000, Windows XP, Windows Services, control-panel app | VS0201DT_T |
| Q&A: Add Images to MDIForm Backgrounds | Karl E. Peterson | VB6, VB5, VB4/32, WinForms, MDIForms, EXE name | VS0201QA_T |
| ASP.NET: Access Contacts Over the Web | Robert Lair | VB.NET, SQL Server 2000, ASP.NET, Windows 2000, Windows XP, Visual Studio .NET, contact manager | VS0201AN_T |
| Web Services: Develop Across Platforms With Web Services | Brian Noyes | XML, VB6, SOAP, Java, Web services, cross-platform development | VS0201WS_T |
| Database Design: Build Flexible Database Tools | Andy Clark | SQL Server 2000, VB6, ADO Extension 2.5, SQL Server 7, database tools, ADOX, DDL | VS0201DD_T |
| Black Belt: Take Exception to Errors | Bill Wagner | C#, error handling, exceptions | VS0201BB_T |
| Build a Universal XML COM Server | Ed Smetak and Jean Caputo | XML, VB6, ADO, DOM, VC++, COM server | VS0201ES_T |
| VS.NET RC1: Appleman Takes a Look Under the Hood | Lee Thé | Dan Appleman, Visual Studio .NET Release Candidate 1 | VS0201LT_T |
| Evaluate N-Tier Designs | Rockford Lhotka | n-tier architecture | VS0201RL_T |
| Editor's Note: Passport Needs Security to Travel | Patrick Meader | Passport, security, privacy | VS0201EN_T |
| Guest Opinion: Make Broadband Ubiquitous | Andrew J. Brust | broadband, DSL, ILECs | VS0201GO_T |
| December 2001 | | | |
| Build a Secure Web Site | Jonny Anderson | VB.NET, C#, XML, IIS, Visual Studio .NET, security, Web sites, SSL, encryption | VS0112JA_T |
| Access the SQL Server Namespace Library | Francesco Balena | SQL Server 2000, VB5, VB6, SQL Server 7, DTS, Enterprise Manager, SQL Namespace | VS0112FB_T |

| Title | Author | Keywords | Locator+ Code |
|---|---|--|---------------|
| Getting Started: Encapsulate Web Forms Code | Stan Schultes | VB.NET, SQL Server 2000, ASP.NET, Windows 2000 Server, SQL Server 7, Access, encapsulation, Web Forms | VS0112GS_T |
| Desktop Developer: Create Windows Services | Billy Hollis | VB.NET, Windows Service | VS0112DT_T |
| Q&A: Spice Up Console Apps | Bill McCarthy and Brian Noyes | VB.NET, C#, XML, VB6, VB5, Console class, DataGrid, Windows API, application launching | VS0112QA_T |
| ASP.NET: Create Page Templates | Jonathan Goodyear | VB.NET, ASP.NET, IIS, .NET Framework SDK, Visual Studio .NET, page templates | VS0112AN_T |
| Web Services: Wrap Legacy Web Apps | Brian Noyes | C#, ASP.NET, legacy apps, .NET Framework, class wrappers | VS0112WS_T |
| Database Design: Identify Data Uniquely | Josef Finsel | SQL Server 2000, SQL Server 7, databases, scalability, data identification | VS0112DD_T |
| Black Belt: Create Customizable UIs With ATL | Mark Schmidt | Visual C++, UIs, ATL, customization, skinning | VS0112BB_T |
| Turbocharge Your HTML Pages | A. Russell Jones | HTML, IE 5.5, fat client, ASP, VBScript, Microsoft Scripting Host | VS0112AJ_T |
| Understand Reference and Value Types | Bill Wagner | C# keywords, class construct, C++, interfaces, structs, classes, .NET Framework | VS0112BW_T |
| A Veteran's Take on .NET | Lee Thé | Gustav Eydelsteyn, ComponentOne, Microsoft's subscription model | VS0112LT_T |
| Editor's Note: Augment Content Delivery | Patrick Meader | Web services, content delivery | VS0112EN_T |
| Reality Check: And the Answers Are ... | Elden Nelson | survey, programmers' careers, accomplishments | VS0112RC_T |
| Guest Opinion: WinForms Kick-Start ... the Web? | Zane Thomas | WinForms, Visual Studio .NET, XML, XSL | VS0112GO_T |
| November 2001 | | | |
| Secure Your Web Site With Passport | Jon Rauschenberger | VB6, Passport SDK, IIS, authentication, HailStorm | VS0111JR_T |
| Read From the NT Event Logs | L.J. Johnson | VB6, VB5, Windows NT, Windows 2000, Windows XP, event logs | VS0111LJ_T |
| Getting Started: Share Data Between Database Formats | Andy Rosebrock | VB6, MDAC, data sharing, data formats | VS0111GS_T |
| Best Practices: Create Maintainable Classes | David J. Roberts | OOP, class maintenance, cohesion, coupling | VS0111BP_T |
| Desktop Developer: Let Me Count the Ways | Karl E. Peterson | VB6, VB5, enumerations | VS0111DT_T |
| Q&A: How's Your Form? | Bill McCarthy, Don Preuninger, and Joe Dour | VB.NET, C#, VB6, VB5, form references, Windows.Forms | VS0111QA_T |
| ASP.NET: Cache and Carry With ASP.NET | Dino Esposito | C#, ASP.NET, caching | VS0111AN_T |
| Web Services: Manage State in Web Services | Bill Wagner | C#, ASP.NET, state management, Web services, session state, Application objects, Session objects | VS0111WS_T |
| Database Design: Solve Tough Queries With the CASE Function | Michael Levy | SQL Server 2000, SQL Server 7, CASE function, Transact-SQL, SQL queries | VS0111DD_T |
| Black Belt: Store Object State Data | Mark J. Collins | XML, VC++, state data, object state, XML nodes | VS0111BB_T |
| Editor's Note: VS.NET Does Windows | Patrick Meader | Web Forms, Web Services, Windows Forms, Visual Studio .NET, ADO.NET, Desktop Developer column | VS0111EN_T |
| Reality Check: No Experience Necessary? | Elden Nelson | experience, industry, learning curve | VS0111RC_T |
| Guest Opinion: The Buck Stops Here | Josh Lane | software quality | VS0111GO_T |
| October 2001 | | | |
| Pointers Add Power and Safety | Robert Teixeira | VB.NET, .NET SDK, pointers, memory manipulation | VS0110RT_T |
| Enhance Performance With ADO Connection Objects | Dimitrios Tsonis | SQL Server 2000, VB6, ADO, Windows NT, Connection objects, scalability, n-tier, MTS | VS0110DI_T |
| Clear Common C# Hurdles | Don Preuninger and Joe Dour | C#, XML, C# tips, common gotchas | VS0110DP_T |
| Getting Started: Build a Project Documenter Utility | Stan Schultes | VB6, VBDoc, project documenter, debugging | VS0110GS_T |
| Best Practices: Don't Overuse Inheritance | Bill Wagner | OOP, inheritance | VS0110BP_T |
| Desktop Developer: Whip Forms Into Shape | Kathleen Dollard | VB.NET, forms, event handlers, Windows.Forms | VS0110DT_T |
| Q&A: Cure Your Dates of Culture Shock | Jonathan Goodyear and Robert Teixeira | VB.NET, C#, SQL Server 2000, ASP.NET, VB6, SQL Server 7, .NET Framework, date formats, debugging, server controls, GETDATE | VS0110QA_T |
| ASP.NET: Easy E-Mail—It's About Time | A. Russell Jones | VB.NET, ASP.NET, SMTP, e-mail | VS0110AN_T |
| C# Explorer: Manage C# Objects | Bill Wagner | C#, Visual Studio .NET, garbage collection, object management, CLR | VS0110CE_T |

| Title | Author | Keywords | Locator+ Code |
|---|----------------------------------|---|---------------|
| Web Services: Develop Interface-Based .NET Web Services | Juval Löwy | C#, Visual Studio .NET, Windows 2000, Windows XP, Web services, interface | VS0110WS_T |
| Database Design: Save Application Preferences | Dianne Siebold | SQL Server 2000, VB6, ADO, INI files, Registry settings, application preferences | VS0110DD_T |
| Black Belt: Develop an ATL Web Service | Bill Wagner | C#, Visual C++, Windows 2000 Server, Web Services creation, ATL | VS0110BB_T |
| Editor's Note: Feedback Shapes Content | Patrick Meader | feedback, Visual Basic Programmer's Journal, magazine content | VS0110EN_T |
| Reality Check: Pop Quiz | Elden Nelson | programmers' preferences, experiences | VS0110RC_T |
| Guest Opinion: Put Rich and Thin Together | Yasser Shohoud | Netscape, networks, think clients, UI, .NET runtime | VS0110GO_T |
| New Technologies in the Enterprise, Special Issue 2001 | | | |
| Create Powerful Web Reports | Roger Jennings | SQL Server, VB6, MSDE, IIS, Windows NT, MDAC, T-SQL, Web reports, templates, stylesheets | VS01ENRJ_T |
| Integrate Apps With BizTalk | Norman T. Gragasin | VB.NET, SQL Server 2000, BizTalk Server, SQL Server 7, Web services, process automation, information exchange | VS01ENNG_T |
| Build a Mobile Application | Bill Block | VB.NET, ASP.NET, Mobile Internet Toolkit, VS .NET, .NET Framework, mobile apps | VS01ENBB_T |
| Program for the Pocket PC | Steve Makofsky and Andy Harding | eMbedded Visual C++, Windows API, Pocket PC, PDA | VS01ENSM_T |
| Desktop Developer: Build Extensible Lookups With Smart Tags | Thomas Robbins | VB6, Smart Tag SDK, extensible lookups, Outlook, Office XP, DLLs | VS01ENDT_T |
| Database Design: Build Enterprise Apps for the Palm | Ward Hitt and Yelena Pavlovskaya | VB6, AppForge, enterprise apps, Palm OS | VS01ENWH_T |
| Editor's Note: Integrating the Enterprise | Patrick Meader | VB, Windows development, enterprise development | VS01ENEN_T |
| Guest Opinion: Power to the People | Bill McCarthy | VB.NET, Bill Gates, Melbourne | VS01ENGO_T |
| September 2001 | | | |
| Write to the NT Event Logs | L.J. Johnson | VB6, VB5, Windows NT, C++, event logs | VS0109LJ_T |
| Install Databases a Better Way | Josef Finsel | SQL Server, VB6, DTS, databases, SQL-DMO, custom installation | VS0109JF_T |
| Target Multiple Devices | Dan Wahlin | C#, ASP.NET, XML, Visual Studio .NET, Openware SDK, device types, device targets, PDAs, WML | VS0109DW_T |
| Getting Started: Build a Task Scheduler | Stan Schultes | VB6, scheduler component, task scheduler, command-line parameters | VS0109GS_T |
| Best Practices: Learn When to Use N-Tier Designs | Rockford Lhotka | SQL Server, ASP.NET, n-tier | VS0109BP_T |
| Desktop Developer: Kill an App Gently | Karl E. Peterson | VB6, VB5, app shutdown | VS0109DT_T |
| Q&A: Extract Color Values Painlessly | Karl E. Peterson | VB6, VB3, VB4, VB5, GetPixel API, color values, borders, timer intervals | VS0109QA_T |
| Expert XML: Generate and Read XML Documents | Yasser Shohoud | VB.NET, .NET Framework, XmlWriter, XmlReader | VS0109EX_T |
| ASP.NET: Automate Text Graphics Creation | A. Russell Jones | VB.NET, ASP.NET, .NET Framework, LabelMaker, text graphics, graphics creation, WebForm | VS0109AN_T |
| C# Explorer: Dive Into Delegates | Steve Lardieri | C#, delegates, function pointer | VS0109CE_T |
| Web Services: Use COM+ Services With .NET Components | Alan Gordon | VB.NET, C#, SQL Server, Windows 2000, ADO.NET, COM+, dynamic registration, class management | VS0109WS_T |
| Database Design: Use Batches for Faster Throughput | John Pearson | SQL Server, Access 2000, batch execution, multiple commands | VS0109DD_T |
| Black Belt: Emulate VB.NET Error Handling | Darin Higgins | VB6, error handling, Try-Catch-Finally | VS0109BB_T |
| Editor's Note: Get the Most Out of Visual Studio | Patrick Meader | VB.NET, Visual Studio .NET, best practices, magazine features | VS0109EN_T |
| Reality Check: Don't Stop With VB.NET: Try C# | Elden Nelson | C#, VB.NET | VS0109RC_T |
| Guest Opinion: The Missing Business Model | Juval Löwy | business model, Web Services | VS0109GO_T |

Fast, Free Product Information

| Advertiser Web site | Page | | | Advertiser Web site | Page |
|---|-----------|----|--------|---|--------|
| IBM www.ibm.com/db2/win | | 71 | | Sequiter Software www.codebase.com | 53 |
| Infragistics www.infragistics.com | C2, 1, C4 | | | Software FX www.softwarefx.com | 2, 3 |
| Amyuni Technologies www.amyuni.com | 48 | | 16 | SourceGear www.sourcegear.com | 32 |
| BEA http://dev2dev.bea.com/tryworkshop | 6 | | 45 | The Imaging Source www.textcontrol.com | 21 |
| ComponentOne www.componentone.com | 4, 8 | | 49 | VSLive! Conference www.vslive.com | 30, 31 |
| Crystal Decisions www.crystaldecisions.com | 13 | | 27 | VSM Online www.fawcette.com/vsm | 24 |
| Data Dynamics www.datadynamics.com | 19, 41 | | 36, 37 | VSM Print, Online & E-Newsletters www.fawcette.com/vsm | 33 |
| ESRI www.esri.com | 29 | | 70 | VSM Salary Survey Online www.fawcette.com/vsm | 67 |
| FarPoint Technologies www.fpint.com | 10 | | 53 | Wise Solutions www.wise.com | 14 |
| GigaSoft www.gigasoft.com | 57 | | 20 | | |
| IBM www.ibm.com/websphere/ebaydev | C3 | | 17 | Product Showcase | 68-70 |

Advertising Sales: Karin Becker, Group Advertising Director 650-833-7168 • Rich Sims, Regional Account Manager (Southwest & Southeast) 650-833-7110 • Mark Romasco, Regional Account Manager (Northeast) 978-779-8989 • Jeff Lorton, Regional Account Manager (Northwest) 650-833-7104 • David Seymour, Inside Sales Manager 650-833-7148 • Katrina Cepeda, Inside Sales Representative 650-833-7109

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions. Sales Department: 913 Emerson Street • Palo Alto, California, USA, 94301-2415 • 650-833-7100 • 800-848-5523

Got an Advanced Degree?

Find out how much it's worth.

2002 Visual Studio Magazine Salary Survey

Exclusively on www.visualstudiomagazine.com

Locator+ code: VS0206SP_T

FTPOnline

Product names herein are the properties of Fawcette Technical Publications, Inc.

**The alternative to mail merge**

- Designer fully integrated with MS Word
- Robust ActiveX component for producing customized documents

More than just mail merge

- Easily control the document layout with VBScript expressions

Access any data source

- Use ADO, ODBC, DAO or even XML to access one or many data sources

Download and more information available at www.xpertdoc.com

To advertise in this Showcase section, please contact Katrina Cepeda
650.833.7109
kcepeda@fawcette.com

Call Today to Reserve Your Space!

Visual Studio
Magazine

BEYOND BUTTONS

Sax SmartUI is a comprehensive suite of user interface elements all based on a single, unified object model.



Sax.net

www.Sax.net • Info@Sax.net • 800-645-3729

Accounting Source Code

Integral Accounting Basic VB is a full featured Accounting System that includes OE, Inventory, AP, AR, GL, Payroll, and Source Code written in VB 6.0. Printed and electronic documentation are also included. To download a demo, please visit www.stfb.com

(Access & SQL Versions also available)

STFB Inc.

PH: (954) 986-8917

<http://www.stfb.com>

Anex Technologies Inc.

Tel: 509-691-2839
www.anextech.com sales@anextech.com

Barcode TrueType Fonts & SDK

Code39, Code128, Code93, CodaBar, ITF, MSI, UPC, EAN, UCC/EAN, FIM., RSS, PostNet, PDF417, DataMatrix, MaxiCode.

Output to direct device context, WMF, EMF, BMP and TIF. Integrate to VB, Access, Crystal Report, C/C++, Delphi, QuickReport, Oracle Report. Free full-featured trial download available from web.

Barcode Equipments

Laser & CCD barcode scanner, image reader, barcode printers, MSR reader, Cash drawer & many more.

OctoGroup Site Compiler

Compile web sites, ASP applications and catalogs into a single EXE file. Produces EXE with integrated ASP-compatible web server. Setup ODBC data sources and external COM components.

WEBSvr/ASP component

ASP-compatible web server in a DLL. Includes SSL, PNG capturing engine, PHP/ISAPI, multithreaded engine and more.



OctoGroup.NET

www.octogroup.net

Visit our site for a free demo

Bar Code ActiveX and DLLs

"Bar codes look fantastic... TAL has the market on this one."
— VB Programmer's Journal

Easily add professional bar coding capabilities to Windows applications in minutes. High-quality device-independent bar codes. Not fonts! Not bitmaps! Perfect for VB, C++, Access, Excel, Word and Internet Explorer. Supports data binding. Prints all major bar code types — including 2-D.

TAL tech

Free demos at www.taltech.com

800•722•6004

CAD & Graphics Libraries for Developers

vectorDraw.com
www.vectordraw.com

In minutes you can add CAD functionality to your programs; read, display, create and manage 2D & 3D drawings.

Simplicity and solidity on VisualStudio 6.0 & .NET, Delphi, C++ Builder & HTML. Fully object oriented components.

DWG DXF EMF WMF VDF VDI TIF BMP GIF JPG

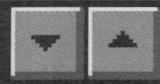
Use this code: **VSM9** to receive a 10% discount!
www.vectordraw.com • info@vdraw.com • +3010 9273 258

ImageKit5 and ResizeKit

The **ImageKit5** adds image-processing capabilities to your applications. Includes Display, Effect, File, Print, Scan, Slideshow, and Thumbnail Controls with over 110 functions.

The **ResizeKit** automatically makes your applications size and resolution independent. No coding necessary. Contains both ActiveX and VCL controls.

NEWTONE Corporation
<http://www.newtone.co.jp>



Need Hands-On Technical Info?

FTPOnline—the new network of technical sites for IT development professionals.

- Downloadable code
- Interviews with industry experts and visionaries
- Overviews of the newest technologies
- Links to other hotspots on the Web
- FREE! Thousands of pages of archives and articles
- Plus, FREE weekly e-newsletters in your area of interest

www.thedotnetmag.com

www.visualstudiomagazine.com

www.javapro.com

www.xml-mag.com



FTPOnline

© 2002, Fawcette Technical Publications, Inc.
All product names herein are the properties of
their respective owners.

www.fawcette.com

SERIAL COMMUNICATIONS

Sax.net provides mission-critical communications when failure is not an option.



8 Cool Controls!

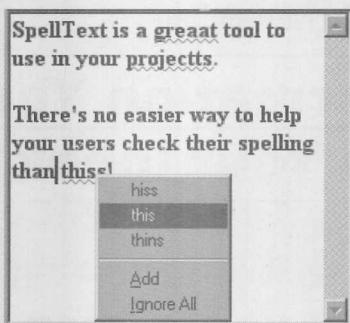
- TE Edit Control Advanced RTF control
- HTML Viewer/Editor Add-on for TE
- SpellTime DLL and dictionary
- RTF Code Generator
- ReportEase Plus report writer engine
- Form Plus
- Rich Text Grid
- ChartPro

Sub Systems, Inc.
TELE: (512)733-2525 FAX: (512)733-2204
Demos: www.subsystems.com

Spell-Checking ActiveX Control

SpellText is a great tool to use in your projects.

There's no easier way to help your users check their spelling than this!



SpellText works like a normal text box, automatically checking spelling.

- Rated A+ by Foxpro Advisor 3/2000
 - Built-in dictionary of 113,000 words
 - Handles any size custom dictionaries
 - Red wavy lines under all misspellings
 - Completely customizable
 - OCX is only 800K in size
 - Includes an Internet-ready CAB file
 - Royalty free, with lifetime free support
- Download an evaluation copy today!

CHADO
ON THE WEB SINCE 1996
<http://www.chado-software.com>

Are You Tired of Paying Royalties?

SKYLINE TOOLS IMAGING PRODUCTS

Royalty Free Products

- **CORPORATE SUITE ACTIVE X Document and Color Imaging**
Color and Document Imaging Package With Source
- **IMAGELIB BARCODE SUITE Barcode Reader**
- **IMAGELIB OMR TOOLKIT Optical Mark Recognition**
- **VIDEOLIB OCX VideoCapture**

DOC-TO-NET

Internet Imaging

Send documents over Internet

Free Trial Downloads:

www.imagelib.com

Phone: (818) 346 4200

email: sales@imagelib.com



Data Compression

Add zip compression to all your Windows programs.



free full evaluation copies
www.innermedia.com

Deliver custom content from automated web sites.



Self-extracting Zip

ReSize

ReSize gives your applications resolution independence. ReSize alters the sizes, font sizes, and positions of controls on Visual Basic forms as they are sized without requiring you to write code. ReSize will give your applications a more professional appearance and allow them to run on a full range of display devices.

Larcom and Young

TELE: (586)771-9597 FAX: (586)771-9564
Free demo: <http://www.lyoung.com>



Follow The Leaders...



...Learn .NET From

*Alison Balter
&
Paul Sheriff*

ASP.NET Basics

Only \$499.95

- 13 Hours of training
- 6 CD's or videos
- 512 page printed course manual
- Exercises, review questions and answers
- Downloadable code

ASP.NET Advanced

Only \$499.95

- 13 Hours of training
- 6 CD's or videos
- 430 page printed course manual
- Review exercises, questions and answers
- Downloadable code

Both ASP courses \$899.95

ASP.NET Architecture

Only \$299.95

- 3 CD's or videos



For a limited time!
all five sets only

\$1,999.95

Save \$300

VB .NET Basics*

Only \$499.95

- 6 CD's or videos
- Printed course manual
- Review exercises
- Downloadable Code
- Ask about advanced

Database Dev. with .NET*

Only \$499.95

- 6 CD's or videos
- Printed course manual
- Review exercises
- Sample Code
- More...

*Available in August

- 30-day money back-guarantee

- Real-world training methodology
- Fits your schedule
- Available on CD or video

- Everything you need to know

- Comes with 5 user license

• Reference AD#VSM



2510 N. University Ave., Provo, UT 84604

To Order: 888.375.4648 • 801.375.5120 • www.BlastThroughLearning.com

CrypKey™

Battle-Proven Software Protection
since 1992

- ▼ new XP support
- ▼ best LAN/WAN support
- ▼ auto Unlock Server

New Version 6.0

Save Time & Money!
Free White Papers

<http://crypkey.com>

Tel: 1-403-258-6274

Schedule, Archive, & Distribute
your Crystal Reports
via Email, Fax, File, & Print

Information Distribution Manager (IDM)

The perfect add-on to enhance the value
and functionality of your current systems

www.goodalesystems.com
or call (310) 768-3968

goodale SYSTEMS, INC.

Introducing .NET Insight Newsletter

**Get the .Net Information You Need—
Delivered to Your Inbox Every Week.**



is going to be huge—and we're making
it easy for you to keep up with it.

Sign up for the *.NET Insight newsletter*, and every week we'll send you timely and relevant information about integrating .Net and technologies like XML, SOAP, SQL Server, Visual Studio .Net, and more. You tell us how much or how little information you want, we wrap it up and send it to you. Every week, absolutely free.

Free newsletter: sign up today!

www.thedotnetmag.com

FIP Online

© 2001, Fawcette Technical Publications, Inc.
All product names herein are the properties of their respective owners.

DB2 software

DB2.

EVERYONE LOVES A WINNER.

IBM

IBM, DB2, Lotus, Tivoli, WebSphere, the e-business logo and e-business is the game. Play to win are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. UNIX is a registered trademark of The Open Group in the United States and other countries. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other company, product and service names may be trademarks or service marks of others. © 2002 IBM Corporation. All rights reserved.

To our customers: Thanks for making IBM the world leader in database software. How did this happen? By providing performance and value across platforms including UNIX®, Windows® and Linux.® The leading application providers: J.D. Edwards, PeopleSoft and SAP all agree. DB2 is how you win the game of e-business. DB2. Part of our software team including Lotus®, Tivoli® and WebSphere®. For more, visit ibm.com/db2/win

e business is the game. Play to win.™

Pick Leaders Who Can Help You

The Visual Basic community has been in turmoil since the announcement of .NET in 2000. Listening to the .NET vs. .NOT debates has prompted me to think about leadership and the qualities leaders share. Basically, leaders are a part of solutions. Leaders help with the hard work of navigating choices, but they don't make the choices for us—choices such as deciding whether .NET is a technology that can help you produce high-quality applications quickly.

Leaders help us find our way through a deluge of ideas and concepts—especially when we're faced with a dramatic debut like that of .NET. Leaders map out a direction through the technology. They don't prescribe a destination. They don't proscribe one either. They lessen anxiety, rather than increase it. They aren't secretive or mysterious. Their communications have a high signal-to-noise ratio, so it's easy to get at the substantive bits of what they have to say.

The best leaders become leaders by going about their business, which looks something like our business. But they take the time to share their experience with us so that our way is clearer. That means leaders have to be willing to go on record. They have to risk airing their opinion. Most of us are opinionated, but not many of us are articulate or care to speak publicly.

Leaders are influential. I've been a developer for 15 years and an MVP for four. I'm lucky to belong to a tight-knit (if sometimes fractious) community. Over the years, I've seen some tremendous leaders rise from the developer ranks. They consistently and significantly influence the direction of a tool. Because of their influence, I have a different experience using programming tools than if they hadn't contributed. I consider particular methodologies and technologies because of their influence.

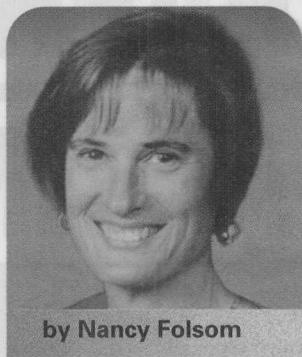
In order to have influence, leaders must be involved in the early stages of development. Even if they aren't early adopters, they must be early users. They have to be willing to suffer frequent, and frequently fatal, bugs. Leaders grit their teeth while reformatting a system between versions. Leaders are writers and editors. They critique documentation that might be, at best, out of date with the software. Leaders are testers. They must have a clear vision of what the tool should be able to do, what it is in fact doing, and how it might be done better. Leaders are masterful debuggers. They perform triage that helps the vendor identify, locate, and either fix or explain the behavior. Leaders try alternative approaches and attempt to identify workarounds. They create simplified scenarios that duplicate the problem. They write

detailed bug reports and otherwise help the development team track down the source of problems. They must be able to communicate clearly. In part, they must be psychologists.

Leaders continually study and absorb information related to their profession. Specifically, they should have experience in more than one language so they can have perspective on how different languages implement computer-science concepts. They are practitioners of good coding practices. It isn't enough to know all the tricks and tics of a language. A significant portion of software development is spent maintaining software. If you've ever inherited an application from another programmer, you know how hard it can be to tease out clever but obscure intent. We look to guidance from leaders not only on how to use technology, but on how best to use it and how to avoid pitfalls. In order to advise us on the best use of a technology, leaders are experienced in several application domains so they can assess how broadly useful a tool might be—not just for their specialty, but for the rest of us.

Given all this intensive work, we expect leaders to be among the people writing books, articles, white papers, and third-party add-ons that make using .NET easier for us. They should volunteer significant amounts of time on the online forums, helping the community understand the technology well enough to make informed choices based on a given situation.

The Tao de Ching says a good leader leads the people where they want to go. That's what I'm looking for as a developer. How about you? **VSM**



by Nancy Folsom

Discuss this opinion in the VB.NET forum on our Web site. Use this Locator+ code: **VS0209GO_D**

The opinions expressed in this editorial are those of the author and not necessarily the opinions of VSM.

Nancy Folsom is a Visual FoxPro developer, Microsoft MVP, owner of Pixel Dust Industries, and author of *Debugging Visual FoxPro Applications* (Hentzenwerke Corp.). Reach her at nancy@pixeldustindustries.com.

WebSphere software

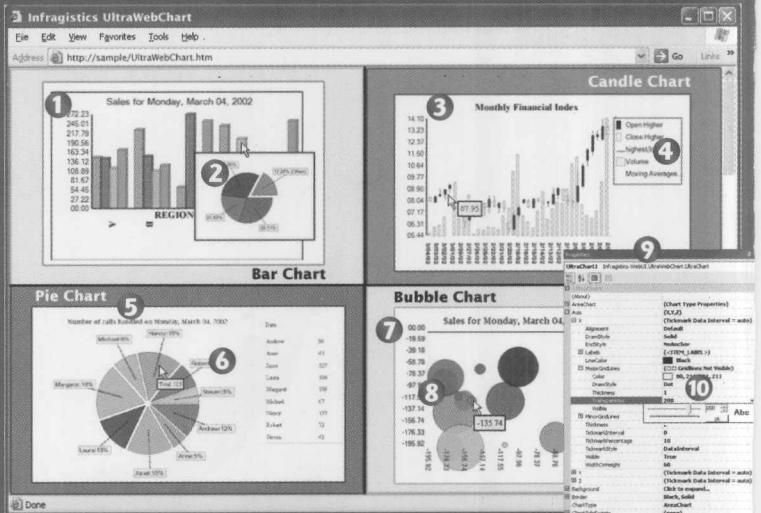
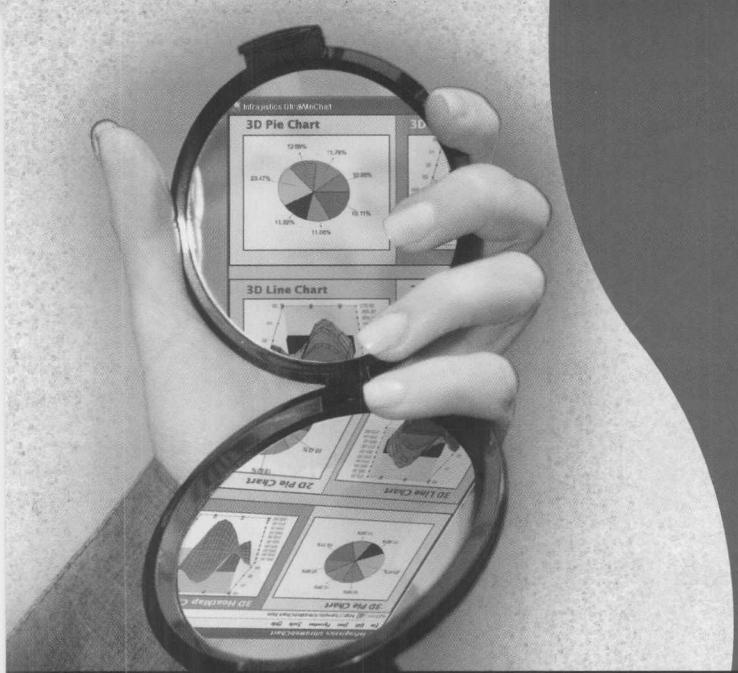
IBM, DB2, Lotus, Tivoli, WebSphere, the e-business logo and e-business is the game, Play to win are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries. eBay is a trademark of eBay Corporation. © 2002 IBM Corporation. All rights reserved.

EBAY IS PLAYING TO WIN.

IBM

business is the game. Play to win.

Winning in the future: eBay has cracked the code of e-business success: keep the customers coming back for more. But how can they more than triple sales by 2005? WebSphere. It's helping eBay slash the time it takes to build and deploy applications, thanks to a tightly integrated runtime infrastructure and development environment. Plus, it's based on J2EE and supports the latest Web services standards, so it can help get your applications out the door faster, too. WebSphere. Part of our winning software team, along with DB2®, Lotus® and Tivoli®. Find out more and get a free trial download of the WebSphere Studio Application Developer at ibm.com/websphere/ebaydev



Mirrored Charting - For Win & Web!

Infragistics - formed by the merger of Sheridan and ProtoView

Infragistics is proud to announce our newest presentation-layer offering - Charting for Windows Forms (Visual Studio.NET) and Web Forms (ASP.NET). UltraWinChart and UltraWebChart share the same object model, and both provide the same rich look-and-feel of a superior client-side control. Present quantitative and categorical information visually and with unprecedented ease. We've provided a comprehensive selection of contextual RAD design-time features, a rich event model, and elegant aesthetic enhancements. Chart data has never before been so exciting, polished and professional.

UltraWinChart™ / UltraWebChart™

- **2D/3D Chart types** - All major chart types are available including Bar, Column, Line, Area, Pie, Scatter, Bubble, HeatMap, HiLow, Financial and Candle Chart.
- **Intelligent Rendering** - Experience maximum productivity and usability through our exclusive IntelliRend™. We've built in every conceivable convenience, intuitive design considerations, and advanced developer control into our Charts. Look for this mark ^{IR}, to identify a feature that saves you time, effort and code!
- **Data bound and non-bound modes** - We provide seamless integration with .NET data sources and containers. Populate your chart from a database or array, using the DataSource property. Internal filtering ensures the data model supplied matches the chart type chosen. ^{IR} Manually exclude/include data columns and rows.
- **Robust Aesthetic Enhancements** - Charting is really about how your data looks. Make your chart visibly superior with anti-aliasing, alpha blending, transparency ^{IR}, background images, an advanced Color Model ^{IR}, and an innovative layered design.
- **Advanced Security**^{WEB} - Our exclusive, non-visual component, SecureImagePipe™, provides ASP.NET session based security for protecting rendered chart data.
- **Event Handling**^{WEB} - Our exclusive ActiveImges™ generates the scripts needed for client and server-side event delegation, seamlessly and accurately.
- **Attention to Detail** - RAD design-time configuration, automatic data scaling ^{IR} & congestion control, data highlighting ^{WIN}, advanced tooltips, fully customizable legends and labels ^{IR}, printing and more!

^{WEB} UltraWebChart only ^{WIN} UltraWinChart only ^{IR} IntelliRend Technology

Copyright 2002 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, UltraWebChart, UltraWinChart, NetAdvantage Suite, SecureImagePipe, IntelliRend, ActiveImage are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

- ① 2D Bar Chart
- ② Advanced ToolTip (image rendered in tooltip)^{IR WEB}
- ③ Candle Chart
- ④ Chart Legend
- ⑤ 2D Pie Chart
- ⑥ Formatted Chart Labels^{IR}
- ⑦ Bubble Chart
- ⑧ Layered Transparency of Chart elements^{IR}
- ⑨ Contextual RAD (Design-time) Properties^{IR}
- ⑩ Transparency Editor^{IR}

UltraWinChart & UltraWebChart are now available in NetAdvantage™ Suite 1.2

For detailed information:
www.infragistics.com
Download Free Trial Version!
Order Online!
or call 800-231-8588

Infragistics™

Component powered infrastructure